AN OBJECTIVE EVALUATION OF FOUR SAR IMAGE SEGMENTATION ALGORITHMS

THESIS

Jason B. Gregga, Captain, USAF

AFIT/GE/ENG/01M-12

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U.S. Government.

AFIT/GE/ENG/01M-12

AN OBJECTIVE EVALUATION OF FOUR SAR IMAGE SEGMENTATION ALGORITHMS

THESIS

Presented to the Faculty

Department of Electrical and Computer Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Electrical Engineering

Jason B. Gregga, B.S.

Captain, USAF

March 2001

AFIT/GE/ENG/01M-12

AN OBJECTIVE EVALUATION OF FOUR SAR IMAGE SEGMENTATION ALGORITHMS

Jason B. Gregga, B.S.
Captain, USAF

Approved:

7 Mar 01

Steven C. Gustafson (Chairman) date

7 MAR 01

Roger L. Claypoole, Jr. (Member) date

7 MAR 01

Eric P. Magee (Member) date

Acknowledgments

I must first express my thanks to the Chief Engineer, without whom nothing was made that has been made. He is my first and greatest teacher. Thanks also go to my wife and daughter, whose personal sacrifice, constant encouragement, and smiling faces have been a source of strength for me. I don't know what I'd do without them.

I am deeply indebted to my committee, Dr. Steven Gustafson, Major Roger Claypoole, and Major Eric Magee. They have been my teachers, my mentors, and my role models.

Thanks also to my sponsor, Dr. Gregory Power, who has always answered my questions, and been deeply interested in my work.

Jason B. Gregga

Table of Contents

List of Figures

List of Tables

AFIT/GE/ENG/01M-12

Abstract

Because of the large number of SAR images the Air Force generates and the dwindling number of available human analysts, automated methods must be developed. A key step towards automated SAR image analysis is image segmentation. There are many segmentation algorithms, but they have not been tested on a common set of images, and there are no standard test methods. This thesis evaluates four SAR image segmentation algorithms by running them on a common set of data and objectively comparing them to each other and to human segmentors. This objective comparison uses a multi-metric approach with a set of master segmentations as ground truth. The metric results are compared to a Human Threshold, which defines the performance of human segmentors compared to the master segmentations. Also, methods that use the multi-metrics to determine the best algorithm are developed. These methods show that of the four algorithms, Statistical Curve Evolution produces the best segmentations; however, none of the algorithms are superior to human segmentors. Thus, with the Human Threshold and Statistical Curve Evolution as benchmarks, this thesis establishes a new and practical framework for testing SAR image segmentation algorithms.

An Objective Evaluation of Four SAR Image Segmentation Algorithms

1 Introduction

This thesis objectively compares four Synthetic Aperture Radar (SAR) image segmentation algorithms. This chapter outlines the problem statement and discusses the thesis goal and organization.

1.1 Problem Statement

The Air Force has listed Information Superiority as one of its core competencies [AFDD-1, 1997:31]. To this end, there are numerous platforms for gathering information. Many of these platforms gather SAR data because it is relatively easy to do so and data collection can be done in any sort of weather [Kuttikkad and Chellappa, 2000]. However, every time a SAR image is generated a human analyst must evaluate it. Because of the increasing number of available SAR images and the decreasing number of available human analysts, the Air Force needs automated SAR image analysis. The first step in this process is automated SAR image segmentation.

It seems fortunate that a large number of image segmentation algorithms are available, as any search can quickly verify. However, this availability leads to the problem addressed in this thesis: how do we make a good decision about which SAR image segmentation algorithm is best? The problem deepens when we realize that the

SAR image data available for testing our segmentation algorithms has little or no associated truth data. The nature of SAR imagery forces this situation. We might be tempted to simply select a scene, gather some SAR data, photograph the same scene from the same angle and distance, and use the photograph for truth data. But SAR images and photographs have different properties. For example, microwave radar returns from an object in one pixel can spill over into another pixel, whereas these effects are much less noticeable in visible light systems. Thus, photographic truth data is not as useful for pixel-level ground truth as we might hope.

However, even if truth data were available and we could state with certainty what the perfect segmentation of a SAR image should be, how would we deal with less than perfect segmentations? What do we do with two different segmentations of an image that are imperfect in two different ways? There are many metrics available that quantify the similarity between two different segmentations of the same image, but as Power has shown [Power, 2001], single metrics do not tell the whole story. Therefore, this thesis uses a multi-metric approach to deal with less than perfect segmentations. In particular, the similarity between segmentations is measured in multiple ways, and a combination of metrics is used to judge the quality of the segmentations. Also, since no truth data is available, the segmentations generated by the algorithms are compared with those generated by humans, which are termed master segmentations and are assumed to be ground truth.

1.2 Thesis Goal

The goal of this thesis is to determine whether any of the algorithms produce segmentations that are good enough to replace human segmentations and to show which of the algorithms is best. Additionally, this thesis develops a novel framework for testing SAR image segmentation algorithms. With the developed framework, objective answers to the questions posed in the Problem Statement are obtained.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 discusses background topics such as SAR imagery, image segmentation, and segmentation metrics and describes the algorithms to be compared. Chapter 3 discusses the methodology for conducting comparison experiments and achieving results, while Chapter 4 presents and discusses the results. Finally, Chapter 5 states conclusions and indicates directions for future research.

2 Background

2.1 SAR Images

The appeal of SAR imaging systems is widespread, and for good reason. Unlike electro-optical imaging systems, SAR can operate in any ambient light conditions, any weather conditions, and can even penetrate some foliage. This power does not come without a price, however. SAR images have inherent speckle. This speckle results from the coherent addition of returned radar signals reflected off multiple scatterers in a scene [Kuttikkad and Chellappa, 2000]. Because the signals travel various distances from the antenna to the scatterer and back to the antenna, their addition is sometimes in-phase (constructive) and sometimes out-of-phase (destructive). Because we cannot yet produce detailed computer simulations of multiple scatterers in a scene, we are forced to model this speckle statistically. Since speckle is induced by the scene and not the radar system, the statistical models represent the scene more than any aspect of the radar system.

There is sufficient speckle in SAR images to make discernment of the edges nearly impossible. In particular, no two humans will segment a SAR image exactly the same way, because no two humans will "see" exactly the same edges in the image. Figure 1 shows a SAR image with edges "seen" by two different humans and indicates that there are few if any agreed upon edges in the image (perhaps the bottom of the target, outlined in lighter gray in Figure 1, qualifies). The difficulty in agreeing upon edge locations is fundamental for SAR image segmentation.

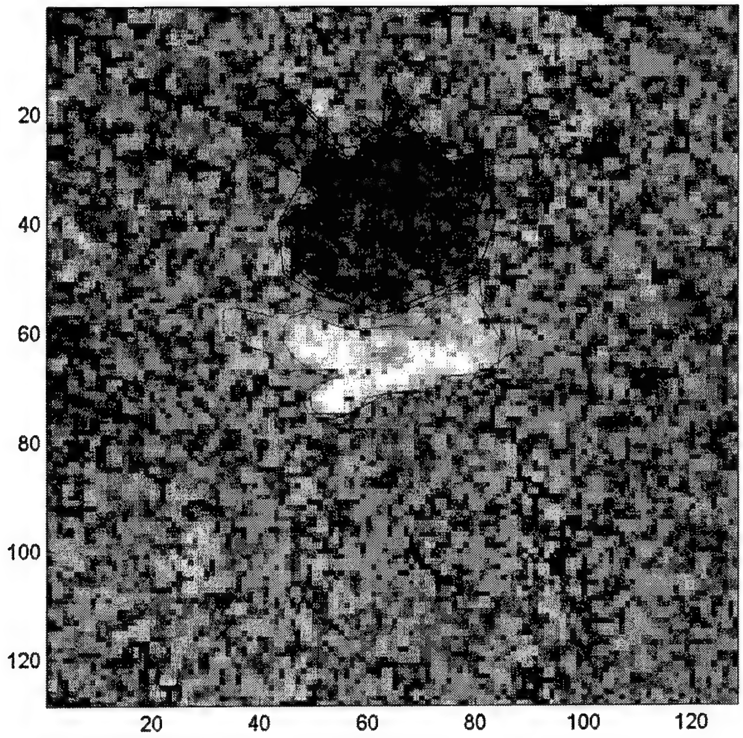

Figure 1—SAR Image of a tank and its shadow and two human segmentations (indicated by thin lines that outline the two objects). The variability of human segmentations is apparent.

The situation is not entirely hopeless, however. Humans have been analyzing and successfully interpreting SAR images since the inception of SAR. Therefore, even though speckle is bothersome, differences in segmentations by humans do not keep us from interpreting the image. Thus image information can certainly be exploited by computer algorithms. We do not expect computer algorithms to agree any more on the edge locations than humans, but we would like them to agree at least as much as humans.

Many probability density functions (PDFs) have been proposed to model SAR data, including Rayleigh, multi-variate complex Gaussian, Weibull, and K PDFs. Weisenseel [Weisenseel et al., 1999] showed that the MSTAR target chips, for example (the data used in this thesis), have magnitudes which correspond closely to the Weibull distribution. Statistical models can be used to perform reliable detection (i.e., determining the class from which a pixel is drawn) on the pixels in an image.

SAR images can be collected any time of day or night. The speckle inherent in SAR makes it difficult to exploit this information, but humans can do so, and thus there must be sufficient information in the image to perform segmentation.

2.2 Image Segmentation

Image segmentation is a somewhat fuzzy concept. However, the general idea is to divide an image into regions which are homogenous internally with respect to some set of properties, and which differ from their neighboring regions with respect to some set of properties. For example, we might want to segment an image that contains a person into two regions: person and not-person. The regions are homogenous in that all of their pixels are in the designated classes. The segmentation process is relatively easy for a photograph and is accomplished unthinkingly by humans.

However, humans bring a lifetime of learning and thinking to the segmentation problem, whereas a computer can do only what it is told, so the problem is significantly more difficult for a computer. Still, there are many successful image segmentation algorithms. The key is to define the properties that are to be homogenous inside regions and different between neighboring regions. If these properties can be calculated by a computer, the rest of the problem is relatively simple. Unfortunately, there is no single property that, for example, defines the class “person.” A person’s face and a person’s clothes look entirely different, yet we are still able to determine that both the clothes and the face are "person." A computer may need many properties to make such distinctions.

Image segmentation can be accomplished in essentially two different ways. One way is to label the class of every pixel in the image. Of the four segmentation algorithms presented in Section 2.5, three of them use this method. Another way is to find the borders (edges) which divide the regions from their neighbors. The fourth algorithm employs this method.

2.3 Segmentation Metrics

This section discusses the metrics used to judge segmentations and the need for more than one metric. The three metrics each measure a different aspect of the segmentation, and thus all three are necessary to make a good judgment about how good the segmentations are.

2.3.1 Percent Pixels Same (PPS)

This metric is the most straightforward of the three, and in some ways, it seems to offer the best measure of a segmentation. Given two binary (1 in the class of interest and 0 everywhere else) segmentation maps of a particular class in the same image, $\hat{S}_1$ and $\hat{S}_2$, we can compare them by simply finding the percentage of pixels that are classified the same way [Power and Awwal, 2000]:

$$m_{PPS} = \frac{\sum_{i,j} \hat{S}_1 \otimes \hat{S}_2}{\max\left\{\sum_{i,j} \hat{S}_1, \sum_{i,j} \hat{S}_2\right\}}, \tag{1}$$

where the $\otimes$ operator is a point-by-point multiplication of the two matrices, and the indices i, j run over the entire matrices. The denominator ensures that $0 \le m_{PPS} \le 1$. If $\hat{S}_1$ is a master segmentation (assumed to be truth), then m_{PPS} is a measure of the quality of $\hat{S}_2$. Note that if $\hat{S}_1 = \hat{S}_2$, then $m_{PPS} = 1$ and $\hat{S}_2$ is declared to be perfect. Also, if $\hat{S}_2$ is 0 everywhere, then $m_{PPS} = 0$ and $\hat{S}_2$ is completely wrong. However, $\hat{S}_2 = 0$ is not the only way in which m_{PPS} can be 0.

If $\hat{S}_2$ has exactly the same shape as $\hat{S}_1$ but is shifted so that it is not found in any of the pixels where the $\hat{S}_1$ shape is located, then $m_{PPS} = 0$, which is not necessarily a desired response. Something right is happening to get the correct shape, but this type of segmentation is deeply flawed, and m_{PPS} does not give all the information necessary to judge $\hat{S}_2$.

2.3.2 Partial Directed Hausdorff (PDH)

For this metric we look at the edges of the segmentations, as defined by a set of vertices [Beauchemin et al., 1998]. Whereas PPS measures the mass of the segmentation that is correct, PDH measures the quantity of edge that is correct. Let $A = \{a_1, a_2, ... a_p\}$ be the set of points which define the edge around the master segmentation and let

$B = \{b_1, b_2, \ldots b_q\}$ be the set of points which define the edge around the segmentation to be tested. The directed Hausdorff distance from A to B is defined as

$$h(A,B) = \max_{a \in A}\left[\min_{b \in B}(\|a-b\|)\right], \tag{2}$$

which is the maximum distance between the points in A and their nearest corresponding neighbors in B. Expanding this equation produces the partial directed Hausdorff distance [Beauchemin et al., 1998], which is the K^{th} ranked distance

$$h_K(A,B) = \underset{a \in A}{K}^{th}\left[\min_{b \in B}(\|a-b\|)\right] \tag{3}$$

for $1 \le K \le p$. We see that the special case for $K = p$ yields the directed Hausdorff distance. Also, we can set $\delta = h_K(A,B)$ and solve for K to find the number of points in A that are within δ of a point in B, which is denoted $K_{A,B}^{\delta}$. Finally, to normalize we divide by the number of points in the set A [Beauchemin et al, 1998]:

$$m_{PDH} = \frac{K_{A,B}^{\delta}}{p}. \tag{4}$$

This metric is most representative of the differences between the segmentations when both edges have the same number of vertices and when the vertices densely sample the edges. Since this metric represents the difference between the edges, if A has many more points than B or vice-versa, the metric may be artificially high or artificially low. We solve this problem by sampling the edges in an equal angular fashion to generate equal numbers of points [Power and Awwal, 2000], as explained further in Section 3.5.

However, this metric alone, just as in PPS, does not adequately measure all possible flawed segmentations. As for the PPS metric, a segmentation that is exactly the right shape, but in a different location in the image, has $m_{PDH} = 0$. Something is clearly wrong with this segmentation, but something is right as well, so 0 is an inappropriate score.

2.3.3 Complex Inner Product (CIP)

This metric measures segmentation shape and is relatively invariant to scale, rotation, and shift [Power and Awwal, 2000]. It is loosely based on correlation, a standard matching method. For example, if we wish to determine whether or not a received one-dimensional signal $y(t)$ is $x(t)$, the one we seek, we simply correlate [Stremler, 1990] the received signal with the reference signal, i.e.,

$$R_{xy}(\tau) = \int_{-\infty}^{\infty} x(t-\tau)y(t)dt \,. \tag{5}$$

However, the shape of the correlation function depends strongly on the two signals, even in the case of a perfect match. What is desired is that the correlation function in the case of a perfect match be $R_{match}(\tau) = \delta(\tau - \tau_0)$, where $\delta(t)$ is the dirac delta function, which, of course, is not possible for correlation except for the autocorrelation of ideal noise. However, this result is obtained for any function by the inverse filter

$$R_{x,y}^{CIP}(\tau) = \int_{-\infty}^{\infty} \frac{Y(f)}{X(f)} e^{j2\pi f\tau} df \,, \tag{6}$$

where $Y(f) \Leftrightarrow y(t)$ and $X(f) \Leftrightarrow x(t)$ via the Fourier transform. Clearly, if $y(t) = x(t - t_0)$, then $R_{x,y}^{CIP}(\tau) = \delta(\tau - t_0)$ for any $x(t)$. However, the inverse filter is inherently unstable, so in practice we use the amplitude-modulated phase-only filter [Awwal et al, 1990] given here in discrete implementation

$$R_{x,y}^{CIP}(k)=\sum_{n=0}^{N}\frac{Y(n)\exp(-j\angle[X(n)])}{|X(n)|+\varepsilon}\exp\left(\frac{j2\pi nk}{N}\right), \tag{7}$$

where $Y(n)\Leftrightarrow y(k)$, $X(n)\Leftrightarrow x(k)$, $|\cdot|$ is the magnitude operator, $\angle[\cdot]$ is the phase operator, N is the number of points in both $Y(n)$ and $X(n)$ (they must be the same), and ε is a small positive number that avoids division by zero, thereby eliminating the inherent instability of inverse filters.

However, this filter yields a function, and we desire a metric which is at most 1, and at least 0. The goal is to measure the "peakiness" of $R_{x,y}^{CIP}(k)$, i.e., to measure how similar it is to the discrete delta function. First, define peakiness as the $\max_k\{|R_{x,y}^{CIP}(k)|\}$ divided by the mean of the rest of the function. The metric is then given by

$$m_{CIP}=1-\frac{1}{\log_{10}\left(\dfrac{(N-1)\max\limits_{k=0}^{N}\left\{|R_{x,y}^{CIP}(k)|\right\}}{\sum\limits_{m=0}^{N}|R_{x,y}^{CIP}(m)|-\max\limits_{k=0}^{N}\left\{|R_{x,y}^{CIP}(k)|\right\}}\right)+1}, \tag{8}$$

which ensures that $0\le m_{CIP}\le 1$, as desired. Equation 8 first measures the peakiness of $R_{x,y}^{CIP}(k)$. For example, if $R_{x,y}^{CIP}(k)=\delta(k-k_0)$ the peak value is one and the mean of the rest of the function is zero, which yields a peakiness of infinity. Then, to compress the range of values, the logarithm base 10 is calculated, giving our example a value of infinity again, so that the whole denominator is equal to infinity. One is added so that the fraction does not yield numbers greater than one. For the example, the result of the fraction is zero. The fraction is then subtracted from one in order to get values from zero to one, which yields a result of $m_{CIP} = 1$ for the example, as desired for a perfect match.

Because Equation 8 is a new formulation of the CIP metric, some justification must be presented. Figure 2 shows plots of m_{CIP} versus peakiness with the logarithm in the denominator and without it. Notice that m_{CIP} is already at approximately 0.9 for a peakiness value of 10 without the logarithm, providing little distinction for peakiness above that value. In contrast, m_{CIP} increases more linearly when the logarithm is employed, providing good distinction for a much wider range of peakiness values.

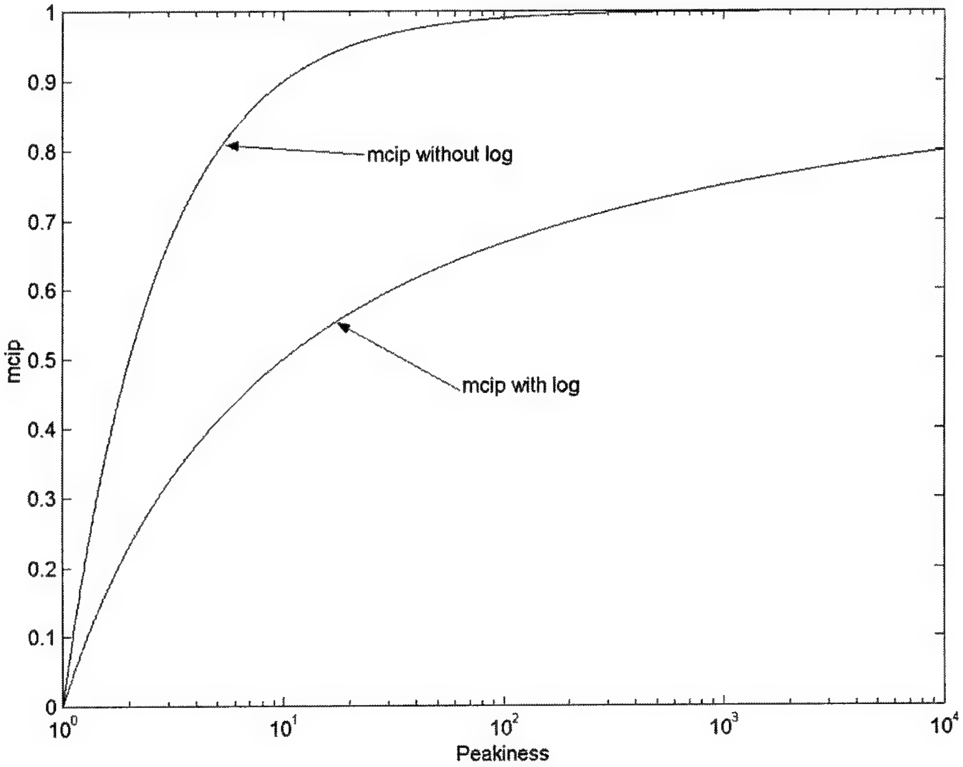

Figure 2—Graph of m_{CIP} versus peakiness showing need for the logarithm in the denominator.

However, the above applies to one-dimensional signals. We can get one-dimensional signals from the sets of edge vertices A and B by simply mapping the x-y coordinates of the points onto the complex plane. Then, $a(k) = a_{kx} + ja_{ky}$ and $b(k) = b_{kx} + jb_{ky}$, which leads to a shape metric for segmentation edge B as compared to the master segmentation edge A:

$$m_{CIP} = 1 - \frac{1}{\log_{10}\left(\frac{(N-1)\max\limits_{k=0}^{N}\left\{ |R_{a,b}^{CIP}| \right\}}{\sum\limits_{m=0}^{N} |R_{a,b}^{CIP}(m)| - \max\limits_{k=0}^{N}\left\{ |R_{a,b}^{CIP}| \right\}} \right) + 1}. \qquad (9)$$

All papers written on the CIP metric use a different calculation method, averaging the frequency domain results:

$$m_{CIP} = \frac{1}{N}\sum_{n=0}^{N} \frac{Y(n)\exp(-j\angle[X(n)])}{|X(n)| + \varepsilon}, \qquad (10)$$

which can yield $m_{CIP} > 1$ if, for example, $|Y(n)| = 2|X(n)|$ (which is still a perfect match since we desire a CIP metric that is invariant to shift, rotation, and scale). However, examples can be conceived where $m_{CIP} > 1$, according to Equation 10, without a match. This troubling result requires a solution and Equation 9 provides the solution used in this thesis. A previous method for dealing with results greater than 1 was to use the phase-only filter (in addition to the amplitude-modulated phase-only filter),

$$R_{x,y}^{CIP}(k) = \sum_{n=0}^{N} \exp(j\angle[Y(n)])\exp(-j\angle[X(n)])\exp\left(\frac{j2\pi nk}{N}\right), \qquad (11)$$

and then taking the minimum between the two [Awwal et al., 1990]. However, this procedure seems to apply the metric unequally, since for some cases it uses the amplitude-modulated phase-only filter, and in others it uses the phase-only filter. Also, the phase-only filter does not use the amplitude information, and it has been shown that the amplitude-modulated phase-only filter provides better discrimination [Awwal et al., 1990]. A better solution is to measure the peakiness as introduced here.

Again, the CIP metric only measures the similarity of the two shapes. There are many possible imperfections in segmentation that this metric will not capture.

2.3.4 Multi-metrics

The need for more than one metric has been emphasized above. In support, Power [Power, 2001] claims that a good evaluation of a segmentation is not possible without more than one metric. He shows that PPS, PDH, and CIP capture enough information to make a good evaluation, as indicated in the test cases shown in Figure 3. Table 1 shows the metric results for these flawed segmentations. The gunless case, (c), is probably the most revealing, although all of these cases show the need for multiple metrics. Since the gun is a relatively small feature (both in number of pixels occupied and number of edge points) PPS and PDH yield relatively good scores. Either of these metrics alone might indicate a very good segmentation, and the combination of them leads us to believe that the segmentation is very good indeed. However, CIP indicates that something is quite wrong. From the three of these metrics together, we surmise that most of the segmentation is correct, but some prominent feature is either missed or added.

Missing from Power's work is a method for using the metrics together. It is not clear how to determine whether some of the flawed segmentations are better than others. Since identifying the best segmentation algorithms is the goal of this thesis, methods for using these three metrics together are outlined in Section 3.7.

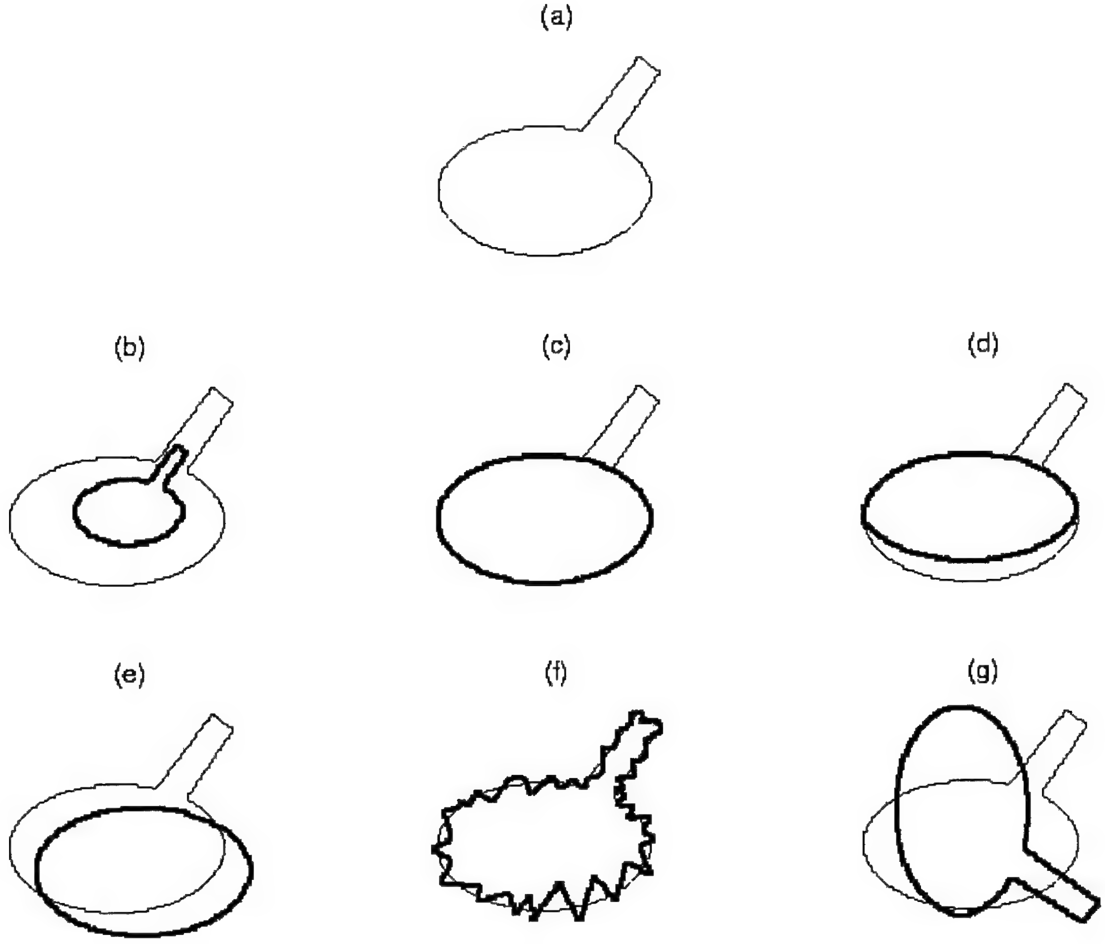

Figure 3—Test cases of flawed segmentations

Table 1

Metric Evaluation of Test Cases

Test Case	PDH	PPS	CIP
a) ideal	1.00	1.00	1.00
b) scaled	0.00	0.26	0.69
c) gunless	0.92	0.90	0.06
d) bottomless/gunless	0.47	0.73	0.07
e) offset gunless	0.02	0.40	0.14
f) noise edges	0.30	0.85	0.38
g) rotated	0.03	0.27	0.77

2.4 Segmentation Evaluation Studies

There is a surprising lack of studies that evaluate image segmentation algorithms. Many segmentation algorithms have been written for SAR and other imagery systems, but most such algorithms have been evaluated by subjective visual inspection (mainly by their authors). The evaluation of SAR image segmentation has been particularly beset by the difficulty of obtaining ground truth. Weisenseel uses percent pixels different to evaluate the performance of his algorithm [Weisenseel et al., 1999]. To do so, he compares his algorithm to hand segmented SAR images and then compares his scores with another SAR image segmentation algorithm. However, he does not use multiple metrics, and it is not clear that the hand segmentations have been produced in any sort of careful way (in contrast with the master segmentations introduced in Section 3.4). The other three algorithms considered in this thesis, much like many existing algorithms, were not evaluated in any objective way.

Hoover et al. [Hoover et al., 1996] suggest a valid study of segmentation algorithms must use multiple metrics, a significant number of real images, and ground truth for those images. Several segmentation algorithms for range imagery (laser range finder and light scanner images) are evaluated, but most of the metrics used are not appropriate for SAR imagery, and the framework for making judgments based on these metrics is complex. Furthermore, pixel-level ground truth data for SAR images is generally not available. Range imagery is much less speckled than SAR imagery, so hand segmentation (used by Hoover et al. for ground truth) is likely to yield good results. No comprehensive study of SAR image segmentation algorithms meeting the criteria outlined by Hoover et al. has yet been performed. Consequently, the framework for evaluating SAR image segmentation has not yet been defined. Power [Power, 2001] establishes that multiple metrics are required and gives appropriate metrics to use, but criteria for making judgments are missing. However, it is hinted that the performance of an algorithm should be compared to the performance of a human, a technique which is expanded upon through the use of what is called the Human Threshold (Section 3.6).

2.5 The Algorithms

This section reviews the operation of the four algorithms, and suggests improvements. The first algorithm is from the University of Minnesota, and is referred to in this thesis as the UM algorithm. The second algorithm is from Boston University, and is referred to here as the BU algorithm. The third and fourth algorithms are both from the Massachusetts Institute of Technology, so they are referred to here by their primary authors initials, AK, and AT, respectively. These algorithms were chosen as samples of recent SAR image segmentation algorithms by the sponsor of this work, AFRL/SNAT.

2.5.1 Knowledge-Based Segmentation (UM)

The main idea of this algorithm [Haker et al.] is to calculate the likelihood that a pixel belongs to each segmentation class and then to choose the class which corresponds to the maximum a posteriori (MAP) likelihood. This algorithm requires a priori information about the probability of each class and assumes a uniform probability for each class in each pixel location. Unfortunately, because of speckle in SAR images, simply choosing the class which corresponds to the MAP likelihood yields a speckled segmentation, which is undesirable. The algorithm combats this difficulty by anisotropically smoothing the posterior probabilities. This complicated smoothing operation is based on the partial differential equation for affine heat flow and has been used for MRI—another imaging system which has inherent speckle. The algorithm is based on the assumption that the pixel intensity values (SAR magnitude data) are normally distributed, which is not necessarily valid. SAR magnitude pixel intensities have been shown to match quite closely to Weibull distributions [Weisenseel et al., 1999]. However, the algorithm nonetheless achieves acceptable results.

2.5.1.1 How it works

Let $\mathbf{I}$ be the matrix of smoothed pixel intensities (this algorithm smoothes the original intensities slightly before performing the statistical calculations) that make up the SAR magnitude image, and let S be the matrix of class values that truly segment the image (i.e., $\mathbf{S}_{m,n} \in \{1,2,...N\}$ where N is the number of classes present in the image).

Given that the pixel intensities for each class are distributed normally, the conditional probability of pixel intensity given the class is

$$P\left(x \mid S_{m,n}=c\right)=\left(2\pi\sigma_c^2\right)^{-\frac{1}{2}}\exp\left(-\frac{\left(x-\mu_c\right)^2}{2\sigma_c^2}\right), \tag{12}$$

where μ_c and σ_c^2 are the mean and variance of class c, respectively, and x is the sample realization of the random variable $\mathbf{I}_{m,n}$. These parameters can be estimated from pre-segmented images. A simple application of Bayes' rule yields the desired posterior probability

$$P\left(\mathbf{S}_{m,n}=c \mid \mathbf{I}_{m,n}=\hat{x}\right)=\frac{P\left(\hat{x} \mid \mathbf{S}_{m,n}=c\right)P\left(\mathbf{S}_{m,n}=c\right)}{\sum_{i=1}^{N}P\left(\hat{x} \mid \mathbf{S}_{m,n}=i\right)P\left(\mathbf{S}_{m,n}=i\right)}, \tag{13}$$

where $\hat{x}$ is the actual pixel intensity observed in the image at location (m,n). In practice we ignore the denominator, since it is a constant with respect to c. Now let $\mathbf{P}^c$ be the matrix of posterior probabilities for class c. Smoothing is now accomplished with the affine heat flow partial differential equation (details are omitted here because they are not essential to the operation of the algorithm, although this smoothing is a unique aspect of this algorithm). The final segmentation estimate is

$$\hat{\mathbf{S}}_{m,n}=\arg\left\{\max_{c=1}^{N}\left(\overline{\mathbf{P}}^{c}{}_{m,n}\right)\right\}, \tag{14}$$

where $\overline{\mathbf{P}}^c$ is the smoothed $\mathbf{P}^c$.

2.5.1.2 Improvements

An improvement that the authors made to the algorithm was running it on several target chips consecutively and using the smoothed posteriors as new priors for the next image. This is a powerful idea, but the real world does not deliver SAR images of the same exact area in sequence (since SAR systems do not operate as video). However, it is possible to run this algorithm on the same image iteratively to learn the priors, which is an improvement implemented in this thesis. To implement this, the algorithm simply takes the smoothed posterior probabilities and uses them as the prior probabilities to perform the calculation again. The number of iterations is called "relooks;" see Section 3.2.1 for more information. It is apparent that this modification to the algorithm offers some improvement since the best performing parameterization of the UM algorithm (see Section 3.2.1) uses six relooks (as opposed to only one if it offered no improvement) but a comprehensive study of the improvement offered by this modification was beyond the scope of this thesis.

2.5.2 Markov Random Field Segmentation (BU)

This algorithm is also heavily based in statistics [Weisenseel et al., 1999]. In fact, it is somewhat more rigorous than the UM algorithm, but its approach to the speckle problem is different. Instead of simply finding $\mathbf{P}^c$ according to the Bayesian solution and then using some sort of smoothing, this algorithm uses Markov Random Fields (MRF) to mitigate the speckle by enforcing the MRF on the prior probabilities. It does so by essentially penalizing (i.e., assigning lower probability to) differences between classes of neighboring pixels in some fashion. This procedure gives a more rigorous statistical solution which has the potential to include more prior information (e.g., target pixels should be in front of shadow pixels) in the form of anisotropic penalties. Unfortunately, the authors are unable to realize this potential because of technical difficulties in finding the solution. Because the priors vary as a function of the classes of the pixel neighbors, the entire image must be segmented at once by solving the resulting simultaneous non-linear equations, which mandate some sort of non-linear optimization method such as gradient descent or simulated annealing. This optimization is either prohibitively slow or less than optimal (if steps are taken to reduce the complexity and arrive at a solution in a timely manner). One of the steps the authors take to reduce complexity is to make the penalties isotropic. This simplification allows an extension to a continuous mapping, which provides a mathematically simpler solution to the simultaneous non-linear equations (Quasi-Newton optimization). The results are then in the form $\mathbf{P}^c$ and the final segmentation is achieved in the same way as in Equation 14 above.

2.5.2.1 How it works

This section discusses the isotropic solution, since this is the form of the algorithm tested here. As before, let $\mathbf{I}$ be the matrix of pixel intensities (this time with no initial smoothing), N be the number of classes present, and $\mathbf{S}$ be the true segmentation matrix we wish to estimate. To make the notation less cumbersome, we reshape the matrices $\mathbf{I}$ and $\mathbf{S}$ into vectors $\mathbf{y}$ and $\mathbf{x}$, respectively, so that $\mathbf{I}_{m,n} = y_i$ and $\mathbf{S}_{m,n} = x_i$. The distribution of pixel intensities conditioned on the class is given by the Weibull distribution

$$P(y_i \mid x_i) = \frac{\beta(x_i)}{\alpha(x_i)}\left(\frac{y_i}{\alpha(x_i)}\right)^{\beta(x_i)-1} \exp\left(-\left(\frac{y_i}{\alpha(x_i)}\right)^{\beta(x_i)}\right), \qquad (15)$$

where $\alpha(x_i)$ and $\beta(x_i)$ are the scale and shape parameters respectively for class $x_i \in \{1,2,...N\}$. These parameters are determined experimentally by examining human segmented SAR images and using a maximum likelihood criterion for parameter evaluation. The authors demonstrate the goodness of fit for target, shadow, and background classes, and the distributions match the histograms quite closely. Assuming that all the data pixels are conditionally independent (conditioned on the class of each pixel) the distribution of the entire image is given by $P(\mathbf{y} \mid \mathbf{x}) = \prod_i P(y_i \mid x_i)$. Now the MRF is imposed in the prior and is given the distribution

$$P(\mathbf{x}) = \frac{1}{Z}\exp\left(\frac{a}{2}\sum_i \sum_{j\in\eta_i} (\delta(x_i - x_j) - 1)\right), \qquad (16)$$

where Z is a normalizing constant, a is a constant which controls the sharpness of the distribution (which in turn controls the smoothness of the segmentation), η_i is the set of indices of pixels which border pixel i, and δ is the discrete delta function. The function after the summations penalizes differences in class for neighboring pixels. Notice that if $x_i \neq x_j$ there is a penalty of -1, which assumes both homogeneity (penalties the same throughout the entire image) and isotropy (penalties the same for differences in all neighbors). The set η_i is chosen to be the four neighbors above, below, to the right, and to the left of pixel i.

The authors state without proof that the posterior probability $P(\mathbf{x}\,|\,\mathbf{y})$, which is to be maximized over all possible values of $\mathbf{x}$, is proportional to $\exp(-H(\mathbf{x}\,|\,\mathbf{y}))$, where

$$H(\mathbf{x}\,|\,\mathbf{y}) = \sum_i \left\{ \begin{array}{l} \ln[\beta(x_i)] - \beta(x_i)\ln[\alpha(x_i)] + (\beta(x_i)-1)\ln(y_i) - \left(\dfrac{y_i}{\alpha(x_i)}\right)^{\beta(x_i)} \\ -a\sum\limits_{j\in\eta_i}(\delta(x_i - x_j) - 1) \end{array} \right\}. \quad (17)$$

What is somewhat difficult to see in this equation is the dependency of the class of one pixel on the classes of the pixels surrounding it. This dependency prevents us from performing Bayes' rule on each pixel separately (as in Equation 13 above) and thus requires non-linear optimization. To simplify the optimization, the authors allow each element of the segmentation map x_i to be continuous. However, a simple continuous extension in scalar form presents a distance problem. For example, if there are three classes, class 3 is closer to class 2 than to class 1, since $3-2=1<3-1=2$. An elegant solution is to make each element of the segmentation map a vector of length N. Class 1 is then indicated by the vector $(1,0,0)^T$ and the other classes are similarly unit vectors in the x_ith direction. A further simplification to the penalty function gives a new and simpler prior

$$P(\mathbf{x}) = \frac{1}{Z}\exp\left\{-\frac{a}{2}\sum_i\sum_{j\in\eta_i}\frac{1}{2}(\mathbf{x}_i - \mathbf{x}_j)^T(\mathbf{x}_i - \mathbf{x}_j)\right\}. \quad (18)$$

Using this new prior in Equation 17, the following function must be minimized:

$$H(\mathbf{x} \mid \mathbf{y}) = \sum_i \left\{ \begin{array}{l} \ln[\beta(\mathbf{x}_i)] - \beta(\mathbf{x}_i)\ln[\alpha(\mathbf{x}_i)] + (\beta(\mathbf{x}_i) - 1)\ln(\mathbf{y}_i) - \left(\dfrac{\mathbf{y}_i}{\alpha(\mathbf{x}_i)} \right)^{\beta(\mathbf{x}_i)} \\ + a \displaystyle\sum_{j \in \eta_i} \frac{1}{2} (\mathbf{x}_i - \mathbf{x}_j)^T (\mathbf{x}_i - \mathbf{x}_j) \end{array} \right\}. \quad (19)$$

This function allows the use of a gradient descent algorithm (e.g., Quasi-Newton optimization). The isotropy and homogeneity of the penalty function along with the extension to a continuous vector-valued segmentation map allow the use of a relatively fast non-linear optimization method that is guaranteed to find the minimum because H is now a convex function.

The optimization yields an estimate $\hat{\mathbf{x}}$ for $\mathbf{x}$, which is a vector of vector values. If we reshape it into a matrix $\mathbf{P}$ corresponding to the original image (in the same way we reshaped the original image into a vector), then we obtain a matrix of vector values, each of length N. Let $\mathbf{P}^c$ for all $c \in \{1,2,...N\}$ be the matrix of scalar values located in the cth element of each vector in $\mathbf{P}$. This $\mathbf{P}^c$ is the same as in Section 2.5.1.1, and the algorithm simply uses Equation 14 (without smoothing) to generate a final segmentation. The reason we arrive at the same $\mathbf{P}^c$ is that the cth element of the vectors in $\hat{\mathbf{x}}$ can be thought of as the probability that a pixel came from class c based on the transfer to continuous vectors. This view is convenient and would not have been possible without the conversion.

2.5.2.2 Improvements

There are eight pixels surrounding a single pixel and the inclusion of the pixels on the diagonals might enhance the performance of the isotropic algorithm. This inclusion would also make the solution more complicated, but aside from perhaps increasing the time it would take to get a solution, it should not affect the concavity of H, so this change could be made rather simply, although it was not implemented in this thesis.

Additionally, the MRF penalty function and related probabilities are generated in a somewhat ad-hoc fashion. It should be possible to analyze a set of segmented images and determine even more rigorously some appropriate values for the penalty function. One could estimate the probability that a pixel is from a different class than its neighbor by taking the number of pixels with different neighbors and dividing by the total number of pixels. More detailed analysis could even yield better anisotropic information. These studies were not performed in this thesis, so the MRF penalty function given here is used.

2.5.3 Multi-scale Based Segmentation (AK)

The main idea behind this method of segmentation [Kim and Krim, 1999] is a new model of the original SAR image. This model yields a vector, called the evolution vector, for each pixel of the original image and a simple maximum likelihood detector is then used to determine to which class the pixel belongs. Because the novelty of this method is in the model, and not the rest of the algorithm, this section focuses on the model. The power of the model is its ability to capture texture information in the form of the correlation between pixels of different scale. Note that this model does not use a wavelet transform, but pixels in coarser scales are averages (or just sums) of their offspring pixels in the finer scales.

2.5.3.1 How the model works

Let $\mathbf{Q}_1$ be the matrix of *complex* SAR data. We start with complex data because the phase information is important in generating the scales (adding complex numbers yields much different results than adding magnitudes). The unity subscript denotes the finest scale image, and coarser scale images are denoted with subscripts greater than one. The coarser resolutions are generated by

$$\mathbf{Q}_l(m,n) = \sum_{i=2m}^{2m+1} \sum_{j=2n}^{2n+1} \mathbf{Q}_{l-1}(i,j)\,. \tag{20}$$

The recursion is performed $L-1$ times where L is the total number of scales employed. The recursion is realistic in that the coarser scales are estimates of the actual radar returns the sensor would receive operating at a coarser resolution. The phase information present in the set $\{\mathbf{Q}_l\}_{l=1}^{L}$ is not needed, so we designate a new set of images $\{\mathbf{I}_l\}_{l=1}^{L}$ given by

$$\mathbf{I}_l(m,n) = 20\log(\varepsilon + |\,\mathbf{Q}_l(m,n)\,|)\,, \tag{21}$$

where ε is a small positive number such that the argument of log is greater than 0, and if we take $\varepsilon = 1$ we can constrain the resulting $\mathbf{I}_l(m,n)$ to be greater than or equal to 0, which may be convenient for image viewing purposes. We now have a multi-scale set of images, but we desire a measure of the evolution from one scale to the next. The authors introduce an auto-regressive model. First, define a coarse scale operator ζ_i (which will be helpful in selecting out pixels of coarser scales) as

$$\zeta_i(m,n) = \left(\left\lceil \frac{m}{2^i} \right\rceil, \left\lceil \frac{n}{2^i} \right\rceil \right), \tag{22}$$

where $\lceil . \rceil$ represents the greatest integer function, and i is the number of scales to be traversed. For example, if we wish to find the next coarser scale ancestor of pixel (3,3), we use $\zeta_1(3,3) = \left(\left\lceil \frac{3}{2} \right\rceil, \left\lceil \frac{3}{2} \right\rceil \right) = (1,1)$ which is expected because pixels (2,2); (2,3); (3,2); and (3,3) in the finer scale all share the ancestor (1,1) in the coarser scale (here the indices start at 0), and (1,1) will be the result for any of those indices. The scale auto-regressive model can be written

$$\mathbf{I}_l(m,n) = a_{l,0} + \mathbf{E}_l(m,n) + \sum_{i=1}^{R} a_{l,i} \mathbf{I}_{l+i}(\zeta_i(m,n)), \tag{23}$$

where R is the order of regression and $\mathbf{E}_l(m,n)$ is a zero mean white noise process similar to modeling error. The set $\{a_{l,i}\}_{i=0}^{R}$ is constant with respect to (m,n) and indicate the linear dependencies between different scales. Since this model is used to classify different segmentation classes, we include another parameter c which allows different models for different classes:

$$\mathbf{I}_l(m,n) = a_{l,0,c} + \mathbf{E}_l(m,n) + \sum_{i=1}^{R} a_{l,i,c} \mathbf{I}_{l+i}(\zeta_i(m,n)). \tag{24}$$

The auto-regressive coefficients are now determined by minimizing the energy in $\mathbf{E}_l(m,n)$ using the least squares criterion

$$\mathbf{a}_{l,c} = \arg\left\{ \min_{\mathbf{a}_{l,c}} \left(\sum_{m,n} \left[\mathbf{I}_l(m,n) - a_{l,0,c} - \sum_{i=1}^{R} a_{l,i,c} \mathbf{I}_{l+i}(\zeta_i(m,n)) \right]^2 \right) \right\}, \tag{25}$$

where $\mathbf{a}_{l,c} = (a_{l,0,c}, a_{l,1,c}, \ldots, a_{l,R,c})^T$. This vector characterizes the evolution in scale of an entire image. To obtain a different characterization for every pixel, we simply use a window of pixels around (m,n) and develop the evolution vector for that window as if it were a separate image. If we define a window as the K pixels on every side of pixel (m,n), then

$$\hat{\mathbf{a}}_l(m,n) = \arg\left\{\min_{\mathbf{a}_l}\left(\sum_{i=m-K}^{m+K}\sum_{j=n-K}^{n+K}\left[\mathbf{I}_l(i,j) - a_{l,0} - \sum_{k=1}^{R} a_{l,k}\mathbf{I}_{l+k}(\zeta_k(i,j))\right]^2\right)\right\}, \quad (26)$$

where $\hat{\mathbf{a}}_l(m,n)$ is the estimate of the regression vector for the window centered at (m,n). Finally, we obtain a complete representation of the total linear dependencies of the pixels in a window around pixel (m,n) at scale 1 using

$$\mathbf{y}_{(m,n)} = (\hat{\mathbf{a}}_1(m,n); \hat{\mathbf{a}}_2(m,n); \ldots \hat{\mathbf{a}}_{L-1}(m,n)), \quad (27)$$

which is a vector value for each pixel in the finest scale image. This vector has a multivariate Gaussian distribution for the terrain types that the authors examined, which facilitates analysis. We now take this vector representation of the SAR image and use a detection process to determine the segmentation.

2.5.3.2 Improvements

This particular multi-scale representation may be simple to calculate (averages are readily found), but perhaps more information would be available in a more traditional wavelet transform. Wavelet transforms can be fast and have the ability to capture not only texture but also directional texture. Since there are many different wavelet transforms available, we have additional flexibility, although selection of an optimal wavelet transform is somewhat ad-hoc.

The difficulty of finding regression vectors might be mitigated by using a technique other than auto-regression to generate a vector. Perhaps the correlation throughout the scales can be measured in a simpler way.

Neither of these improvements have been implemented in this thesis.

2.5.4 Curve Evolution Based Segmentation (AT)

This algorithm seeks to maximize the difference between some statistic (which could be vector valued) inside the curve and outside the curve, and it is generally invariant to the initial placement of the curve (a common problem with many curve evolution algorithms) [Yezzi et al.]. This procedure is reminiscent of the Fisher linear discriminant, which seeks to optimally separate two classes based on their means and variances. The AT algorithm is attractive because it avoids the windowing problem completely. Its statistic is calculated for all pixels inside the curve and for all pixels outside the curve. The window issue never arises because the curve itself is a sort of window. Unfortunately, noise (or speckle) in an image may be problematic, but by adding a penalty for the total length of the curve, speckle effects are dealt with rather nicely.

2.5.4.1 How it works

The workings of this algorithm are steeped in optimization mathematics because the optimal curve must be found, and the only way to do so is to minimize some energy function of the curve. The energy function used is described below, but the details of the optimization are omitted. Suppose that an image has only two classes; images with more than two classes are dealt with by a hierarchical segmentation whereby the algorithm segments the most prominent feature, then the second most, and so on. If we have a statistic that separates the two classes, then we minimize the energy

$$E = -\frac{1}{2} \| u - v \|^2, \tag{28}$$

where u and v are the values of the statistic inside the curve and outside the curve, respectively. As mentioned, the minimization is quite sensitive to speckle, so we add a term which penalizes the arc-length of the curve. The new energy to be minimized is

$$E = -\frac{1}{2} \| u - v \|^2 + \alpha \int_{\bar{C}} ds, \tag{29}$$

where α is a constant which controls the amount of penalty and $\bar{C}$ is the curve. Although this penalty keeps the optimization from homing in on small areas of speckle to achieve minor decreases in energy, the tradeoff (which can be controlled by α) is that it may round off corners. Note that the statistic used affects the optimization equation, since the gradient of the statistic will be a factor.

2.5.4.2 Improvements

There is not much to improve in this algorithm because it is highly flexible in that it can be used with any statistic, as long as the statistic discriminates between the classes. However, non-linear optimization makes it difficult to readily change statistics, since most non-linear optimizations require the gradient, which may not be easy to determine.

3 Methodology

This chapter describes how the tests were conducted. The goal was to perform the tests fairly and objectively. Differences in the algorithms sometimes prevented perfect fairness, but lack of fairness was mitigated as much as possible. This chapter discusses the data used for algorithm testing, the parameters used for each algorithm, the master segmentations used for comparisons, calculation of the metrics, calculation of the Human Threshold, and determination of the best algorithm.

3.1 The Data Set

Large clutter scenes from the publicly available Moving and Stationary Target Acquisition and Recognition (MSTAR) data set [DARPA] were originally selected. These large clutter scenes are representative of the sort of images that the Air Force needs to have segmented automatically (because they are the kinds of scenes encountered in the real world). But evaluation depends on having good human segmentations, which presented a series of problems. First, it was extremely difficult and time consuming to segment out such large scenes (it could take a day to segment out just one class in a scene). Second, because it takes so long to segment out even one scene, only four or five could be made part of the test, which makes the test less meaningful because a sample average of scores over a number of images is obviously more meaningful with a larger sample. Third, large clutter scenes contain many different classes such as roads, shadows, trees, grass, buildings, etc. (some of these classes are nearly indistinguishable, e.g., roads and shadow), and with five or more classes, statistical separation is likely to be smaller than with only three classes. Fourth, the size of the scenes requires large computer resources: as with any image processing algorithm, bigger images require more memory and run time.

For the above reasons, this thesis uses MSTAR target chips, which are small (128x128 pixels) and thus do not take excessively long to run. They are composed simply of target, shadow, and background, with only one target and one shadow, which makes them extremely simple to segment for humans. Instead of days each image takes a human no more than ten or fifteen minutes to segment, which means that a larger number of images can be used for the test, and in turn makes the test more meaningful. Twenty images were used; all are real SAR images of a T72 tank taken from the same elevation but different azimuths. The images are shown in Figures 15-34 in the Appendix.

The raw SAR magnitude data has a relatively low contrast. Therefore, for viewing purposes, the magnitudes to the ¼ power are used here, i.e., $i_d = i^{1/4}$ where i is the raw SAR magnitude intensity of a pixel and i_d is the displayed intensity, which gave better contrast. This transformation also increased the separation between the probability density functions of the three classes under the Gaussian assumption, so the UM algorithm received these display intensities as opposed to the raw SAR magnitude intensities. Figures 3 and 4 show the probability density functions for the raw magnitudes and for the display magnitudes under the Gaussian assumption. Of course, since the ¼ power operation is not linear, we would not expect the display intensities to remain Gaussian. However, since the raw intensities are not Gaussian initially [Weisenseel et al, 1999], no problem is introduced.

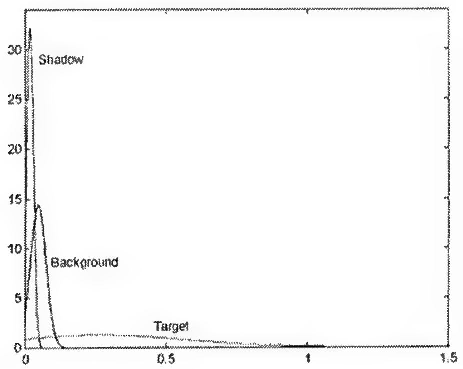

Figure 4—Raw data PDFs under the Gaussian assumption

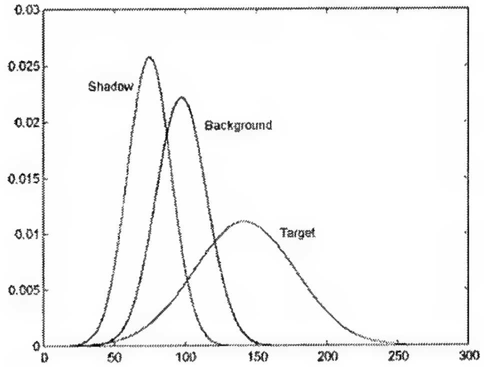

Figure 5—1/4 power PDFs under Gaussian assumption

The BU algorithm received the raw magnitude data because the Weibull parameters that the authors calculated are based on raw magnitude data. Thus using the display data would require recalculation of the Weibull parameters, and it is not clear that the probability density functions would remain Weibull functions. Since this algorithm is based on a more rigorous statistical model of the data, the data should not be changed.

The AK algorithm also received the raw data. It uses phase and magnitude data to calculate the multiple scales, so it would be inappropriate to use the ¼ power in this instance also. But this algorithm does discard the phase data after all the scales are generated and then uses the decibel form of the magnitudes. Experiments using the ¼ power magnitudes instead of the decibel magnitudes were not conducted but might achieve improved results.

The AT algorithm received the ¼ power intensities for the same reasons as the UM algorithm, i.e., since the ¼ power intensities separate the means of the classes better, it should produce better results.

The different magnitude data given to the algorithms does not spoil the objectivity of the comparison. In a sense, using the ¼ power intensities is an improvement, which was implemented in the algorithms for which it was appropriate (UM, AT) and not implemented for the others (BU, AK) since it was not appropriate for them.

3.2 Algorithm Parameters

Each of the algorithms requires parameters. For example, UM requires the mean and variance of each class in addition to parameters such as the number of initial smoothings, number of posterior smoothings, etc. Some of these parameters are very straightforward, such as mean and variance of the pixel intensities in each class, which were estimated from the data and input into the algorithm's code. Other parameters are ad-hoc and are therefore not straightforward, so a pseudo-boot-strapping approach was used. In this approach, the algorithm was first run several times with different ad-hoc parameters until the best-looking (to my eye) segmentations were found. Then a range of these ad-hoc parameters around what looked best was selected. Finally, each parameter in the range was treated as a separate algorithm—that is, all of them were allowed to compete, hence the term "pseudo-boot-strapping". In this way, the process was relatively unbiased.

What follows is a description of the parameter-based procedures for each of the algorithms.

3.2.1 UM Parameters

As described briefly above, the UM algorithm requires the mean and variance of each class, since it is a Bayesian detector that uses a Gaussian assumption for the pixel intensities in each class. To obtain the required values, a binary mask for each class and for each image was created from the master segmentations. These masks each had a value of 1 for every pixel in the class of interest and a value of 0 for every pixel outside the class of interest (for three classes and twenty images, 60 masks were required). It was then a trivial matter to calculate the sample mean and sample variance for each class in this set of images:

$$\mu_c = \frac{\sum_{i,j,k} \mathbf{I} \otimes \mathbf{M}_c}{\sum_{i,j,k} \mathbf{M}_c}, \tag{30}$$

$$\sigma_c^2 = \frac{\sum_{i,j,k} (\mathbf{I} - \mu_c) \otimes (\mathbf{I} - \mu_c) \otimes \mathbf{M}_c}{\sum_{i,j,k} \mathbf{M}_c}, \tag{31}$$

where μ_c and σ_c^2 are the sample mean and variance for class c, $\mathbf{I}$ is the three dimensional array formed by stacking the 20 images, the $\otimes$ operator is a point by point multiplication of two arrays, $\mathbf{M}_c$ is the three dimensional array of the 20 binary masks for class c, and the indices i, j, k run over the entire 128x128x20 array.

One might argue that a truer test would be to calculate the sample mean and sample variance from a different set of images. However, since this procedure would not yield a significantly different number (because of the large number of pixels used) and since it would require human segmentations of all of those other images, the 20 test images were used.

In addition to the mean and variance of each class, UM requires several ad-hoc parameters. The first is the number of relooks, which is the number of times that the algorithm calculates and smoothes the posterior probabilities, then uses the posteriors as the next set of priors. For example, if relooks were set to 2, the algorithm would start with priors of 1/3 for every class and at every pixel, calculate and smooth the posterior probabilities (relook number 1), use the new posterior probabilities as priors, again calculate and smooth the posterior probabilities (relook number 2), and use these posteriors to determine the class of each pixel. This parameter is particular to this thesis, since it is a slightly different implementation of the algorithm than that described by Haker et al. See Section 2.5.1.2 for more details.

The next two parameters are somewhat similar to each other, so they are treated together. These parameters are initial smoothings and posterior smoothings, which are the number of times that the smoothing filter (see Section 2.5.1.1) is run on either the image (in the case of the initial smoothings) or on the posteriors (in the case of the posterior smoothings). For example, if initial smoothings is set to 2 and posterior smoothings is set to 8, the algorithm smoothes the initial image, smoothes it again, performs its calculation of the posteriors, and then smoothes the "image" of posteriors for each of the three classes eight times.

The algorithm was run a number of times for these ad-hoc parameters, which were changed every time until what looked (to my eye) to be the best segmentations were found. Then, a range around the set of parameters was used, as listed in Table 2.

Table 2—UM Parameters

Run number	Relooks	Initial Smoothings	Posterior Smoothings
1	1	2	8
2	1	2	12
3	1	4	8
4	1	4	12
5	3	2	8
6	3	2	12
7	3	4	8
8	3	4	12
9	6	2	8
10	6	2	12
11	6	4	8
12	6	4	12
13	10	2	8

3.2.2 BU Parameters

Like UM, BU also requires some statistical parameters that are not ad-hoc. Since BU operates under the Weibull assumption instead of the Gaussian assumption, the calculation of these parameters is more complicated. However, one of the authors provided the parameters that he used, and they were optimized for MSTAR target chips, so these parameters, listed in Table 3, were chosen.

There is only one ad-hoc parameter to adjust for this algorithm: the penalty, which indicates how stiffly the algorithm penalizes differences in classes between neighboring

Table 3—Weibull Parameters for BU algorithm

Class	Scale parameter	Shape parameter
Shadow	0.0202	1.3058
Background	0.0542	1.7655
Target	0.2182	0.9932

pixels. It can be viewed as how much smoothing the algorithm uses, although it is not strictly smoothing. Again, the algorithm was run numerous times using different values for the penalty until the best-looking set of segmentations were found. Then a range of values was chosen and the different parameterizations of the algorithm were allowed to compete. The values for penalty are listed in Table 4.

Table 4—BU Parameters

Run	Penalty
1	5
2	10
3	15
4	20
5	30
6	50

3.2.3 AK Parameters

Because the AK algorithm also uses pixel-by-pixel detection methods to decide pixel class, it too requires some statistical parameters for each class. However, the value at each pixel is transformed into a vector, and the vector is dependent on an ad-hoc parameter, window size. Since the vector changes with changing window size, the statistical parameters for each class also change for each window size, which means that the statistical parameters must be recalculated for every window size. Unfortunately, time constraints prohibited doing so for as large a number of window sizes as desired (since we are dealing with vectors, the calculations become very time consuming). So for this algorithm window sizes of 17 (17x17), 23 (23x23), and 33 (33x33) were used, where each of these window sizes had a vector mean and covariance matrix for each of the three classes.

The smallest window size the authors used in their work was 33. This algorithm was really designed for much larger scenes than 128x128, so it is somewhat dangerous to go much below this size. However, the test images used here are only 128x128, which means that a window size of 33x33 is already larger than one sixteenth of the image, so larger windows use too much of the image. Therefore, half this size, 17, was chosen; however, the segmentations resulting from a window size of 17 were sufficiently poor to warrant consideration of another window size, so a window size of 23 was chosen. In the future, perhaps every odd window size between one and 63 could be evaluated for a more thorough test.

Additionally, the windowing used in the AK algorithm presents another minor problem: pixels near the edge of the image do not have enough neighbors to provide pixels for the entire window. The solution used here is to simply reflect the image about its edges to the extent necessary.

3.2.4 AT Parameters

Because the AT algorithm uses no underlying statistical assumptions, it needs no statistical parameters of any kind, but it is not without ad-hoc parameters. The first and most important of these is α, which is the curvature penalty. Like most of the other ad-hoc parameters, the algorithm was run with different parameters until the best looking segmentations were found, then a range around that parameter was chosen. The second parameter is not really a parameter but a choice of mode. The algorithm can run in local mode, which means that instead of calculating the mean inside the whole contour and outside the whole contour, it only goes inside the contour a few pixels. This mode can help the algorithm in situations where the interior of an object to be segmented is highly irregular (such as the target). When operating in local mode, a third parameter is needed. The band size tells how many pixels away from the contour to use in calculating the mean. Table 5 lists the parameters and modes used here.

Table 5—AT Parameters

Run	Alpha	Local mode	Band size
1	300	NO	-
2	400	NO	-
3	500	NO	-
4	600	NO	-
5	700	NO	-
6	300	YES	1
7	400	YES	1
8	500	YES	1
9	600	YES	1
10	700	YES	1
11	300	YES	2
12	400	YES	2
13	500	YES	2
14	600	YES	2
15	700	YES	2
16	300	YES	3
17	400	YES	3
18	500	YES	3
19	600	YES	3
20	700	YES	3

Additionally, although not a parameter to be changed, the way in which the algorithm was executed is important. Here, a hierarchical version of the code is used, which means that instead of evolving two different contours simultaneously, it first segmented out the target, then blocked out the target pixels with a mask and looked for the shadow. This is the only version of the algorithm that worked. The only problem with the hierarchical version of the code is in the initialization of the contours. In order to run this algorithm in a fully automated way, the image must be seeded with small contours spread equally and covering the entire image. The algorithm failed to segment out the target when this type of initialization was used because a single contour that enclosed the target was required. Thus this initial contour was chosen to be quite large (approximately 60x60 pixels) compared to the size of the image, but this choice still essentially gave the algorithm some information about where the target was located. The algorithm then attempted to segment out the shadow by not looking at any pixel that was enclosed by the target contour. For this half of the algorithm to work, the initial contour for the shadow must not enclose any of the target pixels (otherwise the part of the contour touching the target pixels did not move), so an initial contour that nearly enclosed the shadow and was above the target was chosen. This choice again gave some location information to the algorithm. Although this location information is vague for both the target and the shadow, and although the same initial contours are used for every image, it cannot be totally discounted. Future work should have AT run in a fully automated mode.

3.3 Algorithm Outputs

The outputs must be briefly discussed because they are different for the different algorithms, and therefore they have been treated differently. The first three algorithms

(UM, BU, AK) all produce the same kind of output. The final segmented image is a 128x128 matrix with values of either 1 (shadow), 2 (background), or 3 (target). Since they are pixel-by-pixel classifications, any edge drawn around any particular class (for the purposes of some of the metric calculations—see Section 3.5) follows the pixel edges and therefore looks somewhat less smooth than desired. The AT algorithm does not perform a pixel-by-pixel classification, so its output is different. It gives a contour around the target and another contour around the shadow. This means that it has already drawn the edge, and it does not necessarily have to follow any pixel edges. It is easy to find which pixels are enclosed by the contour, so the PPS metric can still be calculated. But the somewhat less smooth nature of the edges given by the first three algorithms may slightly affect their scores for the PDH and CIP metrics, since they measure shape. The methodology for calculating the metrics will be discussed further in Section 3.5.

The advantage for AT is not unfair because this algorithm performs image segmentation in a fundamentally different way than the other algorithms (i.e., curve evolution). If the difference yields a better result, so be it.

3.4 Master Segmentations

As mentioned in Chapter 1, there is almost no truth data available for these SAR images. The difficulty in objective comparisons lies precisely in this problem. In order to generate meaningful metrics, the segmentations generated by an algorithm must be compared to other segmentations. To this end, I generated a set of master segmentations. All twenty images were slowly and painstakingly segmented by hand, then the segmentations were reviewed and edited. This was done until the segmentations were as good as they could be. An expert in the field of SAR image segmentation (Dr. Gregory Power, who has been looking at MSTAR target chips for many years) reviewed the segmentations and he concurred that they were good.

A more thorough method of generating master segmentations would be to have multiple (on the order of ten or twenty) experts in SAR imagery look at these segmentations. Each expert in turn could edit the segmentations until he felt that they were as good as they could be, and then pass them along to the next expert. Once all of the experts review the segmentations, the process could be repeated continually until all of the experts are satisfied. Alternatively, all the experts could be gathered in one room and the changes could be voted on. Once no more changes are voted, a final vote could determine whether the majority of experts were satisfied with all of the segmentations. Perhaps future work can attain this goal. Because of time limitations, this thesis did not produce such master segmentations (one cannot simply ask a group of experts to participate in such a study and hope that they will all finish their tasks in a timely manner: such a process could take a year or more). Since the composition of the images is so simple, it is likely that the master segmentations used here are close to the truth. Figures 37-56 in the Appendix show the master segmentations of all twenty images. This thesis assumes that the master segmentations may be accepted as truth data.

3.5 Metric Calculation

The calculation of the metrics requires some manipulation of the segmentations. PPS is very straightforward to calculate, and requires little manipulation. Since it is a pixel mass metric, it requires that the segmentations have each pixel labeled as shadow, background, or target. The UM, BU, and AK algorithms already meet this requirement by their nature. The human segmentations and the AT segmentations have contours and do not expressly label each pixel, but this is not a problem because the pixels that have their centers enclosed by the contours can be found (the function ROIPOLY in Matlab's Image Processing Toolbox does this very easily). We can then label the pixels with the proper class. The PPS calculation then proceeds as detailed in Section 2.3.1.

The PDH calculation requires some more manipulation. First, the human segmentations produce single shapes for both the target and the shadow, and PDH needs the same number of shapes in both segmentations it compares. When a segmentation does not produce a single shape, the single largest shape is selected, and the comparison is performed on that shape only. This outcome is also true for the CIP metric, since it too requires the same number of shapes for comparison. Because of the multi-metric approach, the results are not skewed since PPS is looking at all the regions, not just the largest one.

Besides needing the same number of shapes, PDH needs contours with the same number of points in them for fair evaluation. For the UM, BU, and AK algorithms, after choosing the largest region, the contour that encloses the pixels in that region is found. Once the algorithms have found contours, we ensure that every contour has the same number of points (1024 for the sake of completeness) by angular sampling [Power and Awwal, 2000]. We first find the center of mass in the master segmentation, and then, using that same point for all the algorithms, we sample the contour at 1024 different angles (0 to 2π) from the center point. In general, this produces a much denser sampling of the contour than the original number of vertices in the contour.

However, this process of angular sampling is not without pitfalls, i.e., certain shapes are not sampled completely. The problem is that the center of mass may not "see" the entire contour if a shape wraps around itself, as shown in Figure 6. Fortunately, the master segmentations do not contain such irregularities, so at least they will be sampled properly. But for the algorithms that produce these types of shapes, these are grave irregularities, at least in terms of shape, which should be accounted for and scored appropriately. Because of the angular sampling process, they may actually achieve a better score than deserved, since the irregularities will be ignored by the angular sampling process. To date, there is no good way to alleviate this difficulty, although it does not greatly affect the results since these types of irregularities are rare, and PPS still captures them. Once the angular sampling is complete, the PDH metric is calculated based on the two sets of 1024 points as shown in Section 2.3.2, with $\delta = 1.25$.

To calculate the CIP metric, angular sampling is also used to generate the same number of points for both segmentation contours. It is not clear whether angular sampling is required for the CIP metric or not. It may be possible to use equal distances along the curves of both segmentations to get 1024 points. The advantage would be that it would avoid the problem of not "seeing" the entire contour. Further work should determine whether or not it is necessary to use angular sampling.

Because it is assumed that there is no difference in importance between the segmentation of the shadow and the segmentation of the target, the average of the scores for the target and shadow is used. The three scores then are the scores for the particular algorithm on that particular image.

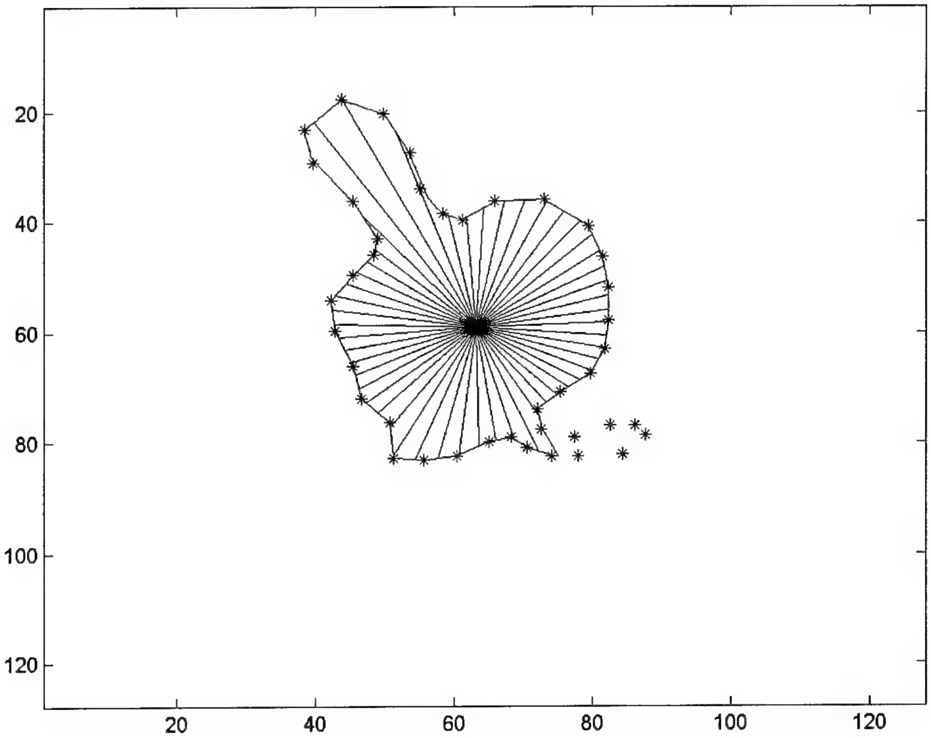

Figure 6—Image showing pitfalls of angular sampling process: the stars are the original vertices, and the lines radiating from the center show examples of the sampling angles. Six vertices are missed.

For every image and algorithm (including every parameter variation), we obtain a set of thee scores: PPS, PDH, and CIP. The scores are then averaged over all twenty images for each particular algorithm. As mentioned earlier, this gives stability to the metrics, since averaging is a smoothing operation.

3.6 Human Threshold Calculation

A key goal is to show whether or not any of these algorithms segment well enough to replace a human segmentor. It is well known that no two humans will segment the same

same image in the same way. In fact, it is nearly impossible for even the same person to segment the same image in the same way more than once. This variability in the way humans segment is key to understanding what level of performance is good enough to replace humans. We are not looking for perfect segmentations, which would be impossible to achieve in any case. Instead we are looking for algorithms that perform at least as well as humans, which suggests that there is some standard, which (if measurable) would be useful. The standard defined here is the Human Threshold (HT), which, following the multi-metric approach, is made up of a set of three scores: PPS, PDH, and CIP. Any algorithm that has all of its scores above the HT is declared good enough to replace a human segmentor.

To calculate the HT, I segmented the twenty images by hand for a second time. Since it had been some time (over a month) since my first painstaking segmentation, it was as if I were a second person looking at the images. The segmentations were done quickly, but carefully. The goal was not to create a second set of master segmentations, but merely to segment the images as a SAR image analyst might. Additionally, another person, trained on this set of images, segmented the images. The metrics were calculated for each of these segmentations, as described in Section 3.5, comparing them with the master segmentations. The lower set (defined by thee-dimensional distance from the ideal—see Section 3.7) is the HT, as shown in Table 6.

Table 6—Human Threshold

PPS	PDH	CIP
0.7923	0.4472	0.5067

Ideally, the HT would be calculated for a much larger number of people, and future work should take this into consideration. A large number (perhaps 100) of SAR image analysts should all segment these images, and their scores should be compared to a set of master segmentations (perhaps generated by the same analysts through the process discussed in Section 3.4). The lowest (also defined by three-dimensional distance from the ideal) set of scores would be the HT. Here, the HT listed in Table 6 is assumed valid.

For an algorithm to be considered as good as or better than the HT, it must have scores equal to or greater than the HT in every metric. This distinction is important: because all of the metrics are needed to completely describe the quality of the segmentation, the HT must be met in every metric.

3.7 Definition of Best Algorithm

The HT may not be met by any of the algorithms. In this case we need some way to judge whether one algorithm produces better segmentations than the other algorithms. If we were using a single metric, this would be a simple matter: the largest number would indicate the best segmentation. However, when using multiple metrics, it is somewhat unclear how to make a judgment. Certainly, since our scale goes from 0 to 1 for each of the three metrics, a score of (1,1,1) is better than a score of (0,0,0), but we need to know what to do for intermediate cases, since our results will likely lie between the two extremes.

We must first define the problem more clearly. The score possibilities can be modeled as a three-dimensional cube. To decide the best scores, we must first decide on surfaces of constant goodness. Then points which lie on surfaces of constant goodness that are closer to ideal can be considered better than points which lie on such surfaces further from ideal. The surfaces may in general be difficult to determine, e.g., they may be application specific, and they may be highly influenced by opinion (one could easily imagine that highly irregular surfaces might be appropriate for some applications). Here, the advantages and disadvantages of three types of regular surfaces are discussed.

The first set of surfaces of constant goodness is generated by constant Euclidean distance from the origin of the cube, i.e.,

$$G_o = \sqrt{m_{PPS}^2 + m_{PDH}^2 + m_{CIP}^2} \,, \tag{32}$$

where G_O is the "goodness" measured from the origin and $m_{PPS}, m_{PDH}, m_{CIP}$ are the metric measurements for PPS, PDH, and CIP, respectively.

This method obviously produces spheres with (0,0,0) as their center and Figure 7 shows one of the spheres. It has a nice normalcy to it, since Euclidean distance is a common standard, but unfortunately, a problem with this method is that, for example, it sets a segmentation score of (1,0,0) equal to a segmentation score of (0.58,0.58,0.58), which means that although a segmentation may do very well in one metric and very badly in the other metrics, it is still considered to be as good as a segmentation that is adequate in all three. This result seems counter to the idea of multi-metrics, which says that all three metrics are needed for complete evaluation. Therefore, points on the sphere of constant goodness (defined by constant Euclidean distance from the origin of the cube) which are close to the sides of the cube should not be as good as points near the middle of the sphere, and thus Euclidean distance from the origin does not measure goodness well.

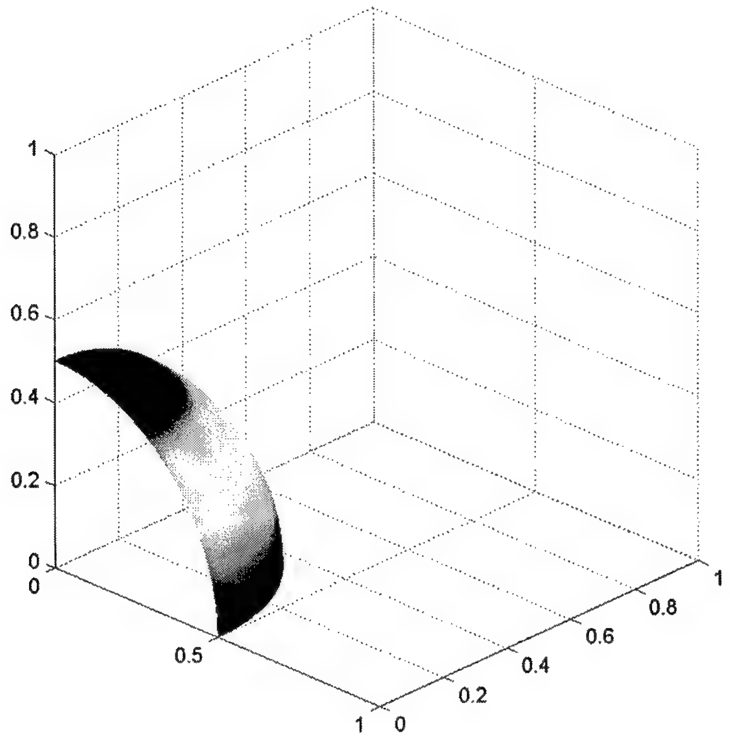

Figure 7—Surface of Constant Goodness using constant Euclidean distance from origin

The second set of surfaces is generated by constant average of the three metrics, i.e.,

$$G_A = \frac{m_{PPS} + m_{PDH} + m_{CIP}}{3}. \tag{33}$$

This method produces planes which are perpendicular to the line which connects the points (0,0,0) and (1,1,1), and one such plane is shown in Figure 8. Obviously, this method also has a niceness to it, since averages are easy to calculate, and it does better than Euclidean distance from the origin at favoring the middle of the cube.

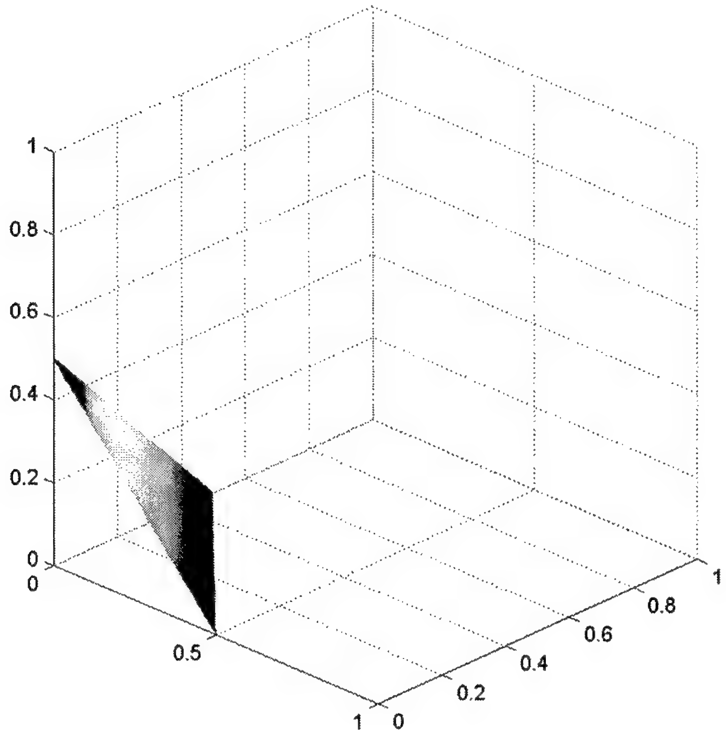

Figure 8—Surface of Constant Goodness using average of three metrics

The third set is generated by constant Euclidean distance from the ideal point, i.e.,

$$G_I = \sqrt{(1 - m_{PPS})^2 + (1 - m_{PDH})^2 + (1 - m_{CIP})^2}\ . \tag{34}$$

This method produces spheres which are centered at (1,1,1), and Figure 9 shows one such sphere. It goes even further than the average method in favoring the middle of the cube, which may be entirely appropriate.

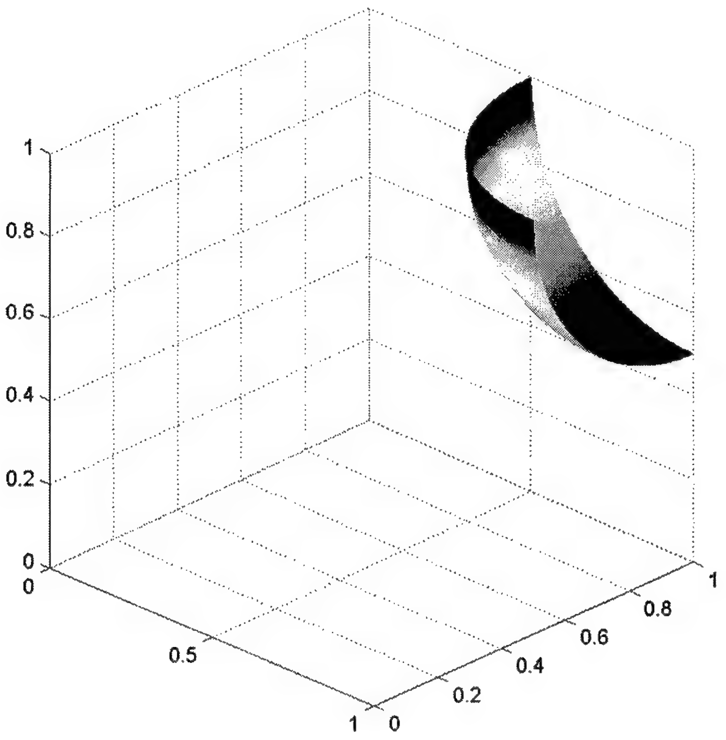

Figure 9—Surface of Constant Goodness using constant Euclidean distance from ideal

There seems to be no clear way to determine which of the three methods is optimal (if any). However, since we must objectively choose a best algorithm, we must choose a method for reducing the dimensionality of the multiple metrics from three to one. I advocate using Euclidean distance from ideal simply because (in my opinion) the middle of the cube should be highly favored, and I do not believe that the average goes far enough, but certainly one could make a case for using the average instead of Euclidean distance from ideal.

However, any projection into a lower dimensional space necessarily reduces the information, if that information truly exhibits three dimensions. This reduction is most expressly realized in the fact that one segmentation score may be called "better" than the

HT under one or many of these methods of judgment, but still not meet the HT in every category and thus not be called good enough to replace human segmentors. The temptation thus exists to declare as good enough to replace human segmentors any segmentation that is in a surface of constant goodness that is closer to ideal than the surface of constant goodness for the HT. However, it is important to remember that information is lost in the reduction of dimensionality. Therefore, if the question is whether or not a segmentation algorithm is good enough to replace a human segmentor, the HT must be the only guide. However, if the question is whether one algorithm is better than another, this thesis uses G_I to answer.

The above argument brings another concern: if G_I is a good enough measure to determine which algorithm is best, and we call the HT one of the algorithms, then it should also be a good enough measure to determine whether an algorithm is better than the HT. However, this statement ignores the fact that the HT is not simply an algorithm: it is a threshold to be crossed. Thus, comparing an algorithm to the HT through surfaces of constant goodness is inappropriate, because we care where on the surface of constant goodness the HT lies, as opposed to comparing two algorithms, when we have no such concern.

3.8 Summary

In summary, each parameter set of each algorithm is treated as a separate algorithm. After performing the segmentations, the metrics are calculated, averages of target and shadow metrics in a single image are found, and then averages of the scores over all twenty images are found. This procedure produces a set of three scores for each parameterization of each algorithm. Also, the HT is computed, and any algorithm that meets or exceeds the HT is deemed good enough to replace human segmentors. If none of the algorithms cross the HT, then one is shown to be best using G_I.

4 Results

4.1 Sample Segmentations

As a sample of the performance of the segmentation algorithms, this section shows the image hb03796 (Figure 10), its master segmentation (Figure 11), and the segmentations produced by the best parameterizations of each of the four algorithms, along with the metric scores for the segmentation (Figure 12-Figure 15). Note that although some scores may be high enough for all three metrics to cross the HT, we are concerned with the average over all twenty images. All scores are averages of the target and shadow scores. More segmentations of this image are provided in the Appendix.

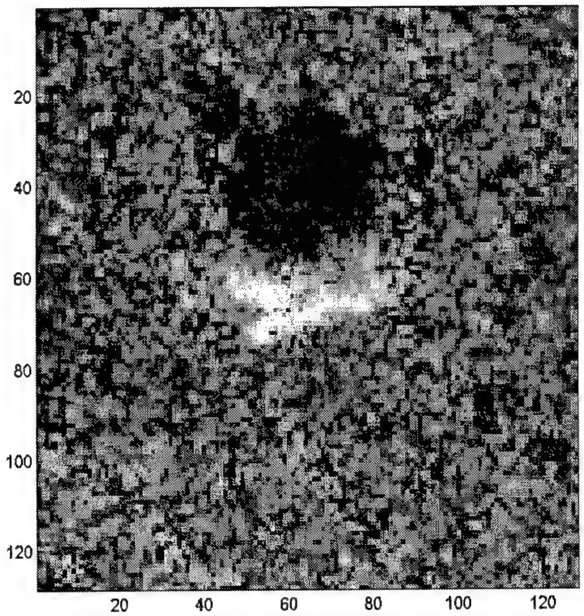

Figure 10—SAR image hb03796, which is a T72 tank (bright spot in the middle) with its shadow (darker spot above the tank).

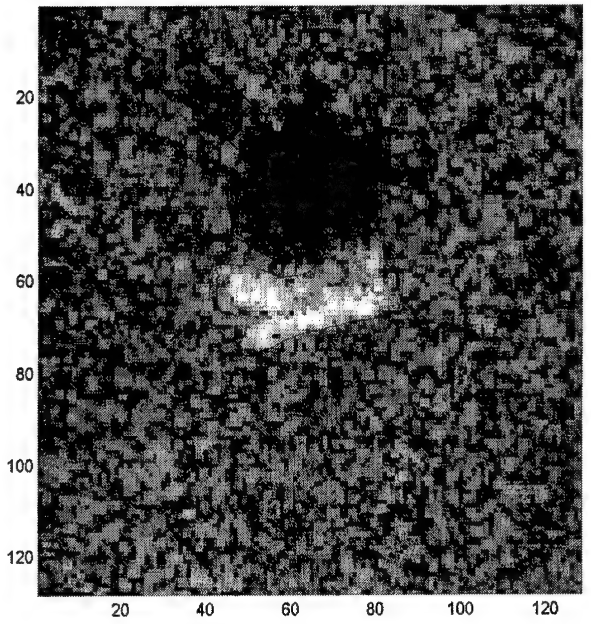

Figure 11—Master segmentation of hb03796

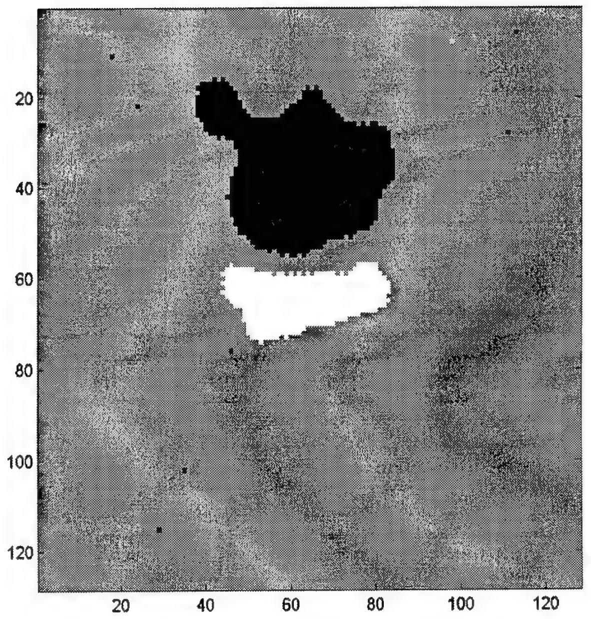

Figure 12—Best UM segmentation of hb03796: PPS=0.7888, PDH=0.6313, CIP=0.5413

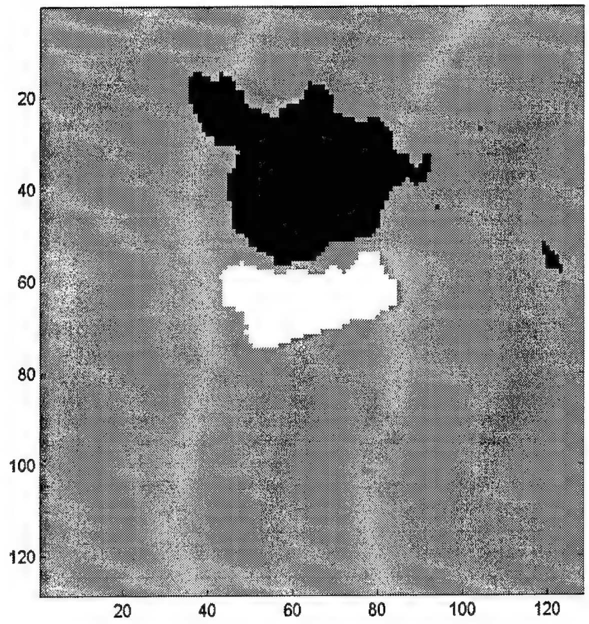

Figure 13—Best BU segmentation of hb03796: PPS=0.8047, PDH=0.6436, CIP=0.4686

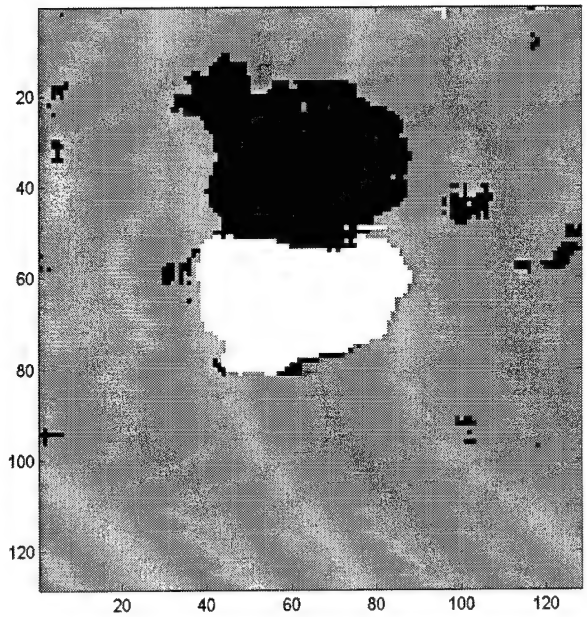

Figure 14—Best AK segmentation of hb03796: PPS=0.5545, PDH=0.1270, CIP=0.5110

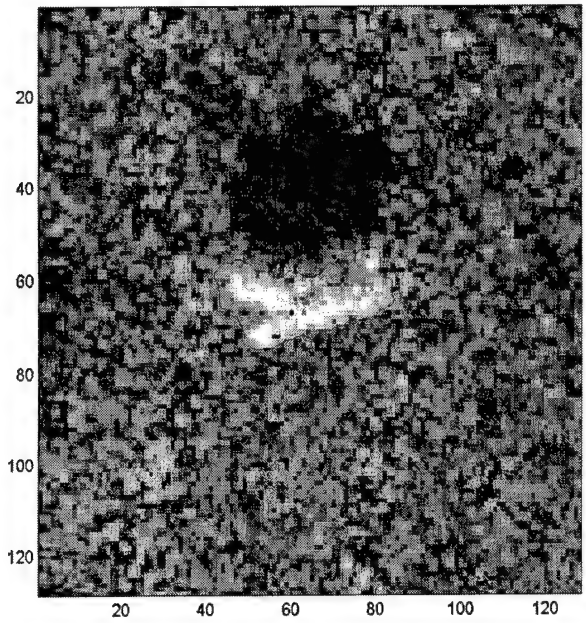

Figure 15—Best AT segmentation of hb03796: PPS=0.8523, PDH=0.6987, CIP=0.5068

4.2 Metric Results

Table 7 shows the metric results for the best parameterization of each of the four algorithms. It also shows whether or not each algorithm meets the HT and G_I, G_A, and G_O. None of the algorithms presented in this thesis meet the HT: the AT algorithm comes closest, but falls slightly short on PPS.

These results show that the AT algorithm is superior to the other three in all three combinations of the metrics. Additionally, a clear ranking of the algorithms is apparent: AT is first, BU second, UM third, and AK fourth. Clearly, since all three combinations of the metrics produce the same results, we are operating in a region of the cube near enough to the center to make the issues discussed in Section **3.7** irrelevant. The scatter plot in Figure 16 shows the metric results for all parameterizations of the algorithms in

Table 7—Best metric results for each algorithm

All scores are PPS PDH CIP	Average over 20 images	Meets HT?	Distance from ideal (min is best)	Average of three metrics	Distance from origin
Alg/run					
UM11	0.7066 0.4077 0.5488	No	0.8003	0.5543	0.9832
BU1	0.7552 0.4341 0.5377	No	0.7706	0.5757	1.0237
AK1	0.5044 0.1070 0.5009	No	1.1367	0.3708	0.7189
AT6	0.7758 0.4917 0.5572	No	0.7104	0.6082	1.0743

the three dimensional cube defined by the metric space. More information can be obtained by looking at the metrics in three dimensions. The AK algorithm is clearly behind the others, due mainly to its very low scores in PPS and PDH. However, its CIP score, while still below the others, is relatively good, which leads to the conclusion that the AK algorithm produces shapes that are pretty good, but are badly scaled or shifted in some way. The BU and UM algorithms offer roughly similar performance, with BU leading in PPS and PDH, but UM leading in CIP. Thus, UM produces segmentations with better shape, but BU produces segmentations with more accurate scale and shift. Perhaps for some applications, one might choose UM over BU, if shape is more important. The AT algorithm performs better than the others in all three metrics, and thus produces superior segmentations in every way measured here.

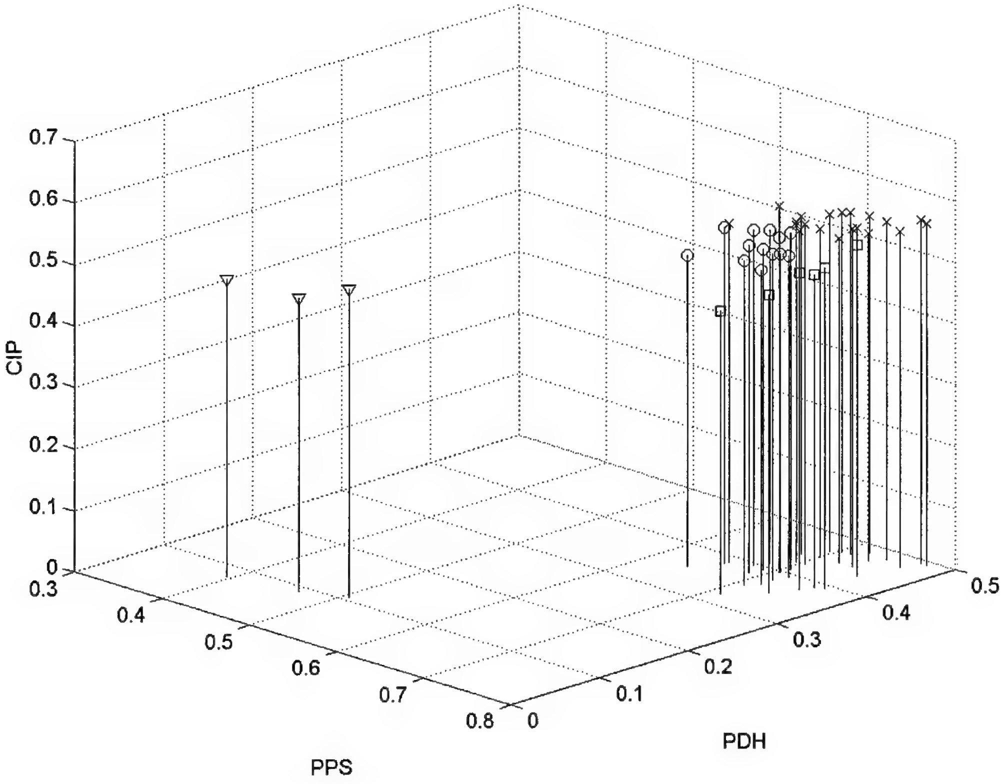

Figure 16—Metric results in 3D metric space: Circle=UM, Square=BU, Triangle=AK, X=AT

5 Conclusions

5.1 Human Threshold Conclusions

Unfortunately, we cannot say that any of the four tested algorithms are good enough to replace human segmentations. However, the results do not show that these algorithms are absolutely not good enough (because of limitations to the scope of the experiments). We can only say with certainty that none of the algorithms are good enough on the employed data set to exceed a particular Human Threshold according to a particular set of master segmentations. We can also point to experiments of the type performed as models for future experiments. A larger data set (i.e., more images containing more pixels) more representative of SAR data collected in the real world, plus more widely agreed upon master segmentations and a more thoroughly measured HT, may show more conclusively whether or not any of these algorithms can replace human segmentations.

5.2 Metric Results conclusions

Conclusions based on the metrics alone are less limited. We can say with certainty that the Curve Evolution Based AT algorithm performs better on the employed data set when compared to the employed master segmentations. A larger data set and more widely agreed upon master segmentations would likely confirm this statement. We can also say that if we had to choose one of these algorithms to replace a human now and without any further information, we would select the AT algorithm.

5.3 Future Work

As discussed in Section 3.4, the master segmentations (which must be good enough to replace ground truth) could be improved. For example, many people who are very familiar with SAR imagery could participate in a test. For each image in the test set, one person could segment the image and pass it to another person, who could edit the first segmentation and pass it to yet another person. This process could be repeated until each person segments all the images and further repeated until no more changes are made. The result would constitute the master segmentations and would be as good as a group of humans could generate.

The HT also could be improved. Through a similar process, each of the images could be segmented by the group of SAR imagery people, but this time each person could segment the images alone. Then the metrics could be run on all of these different segmentations, comparing them to the master segmentations. The lowest set of metrics would determine the Human Threshold.

The CIP metric too needs work in the future. The particular formulation used here seems to do a good job, but there may be other solutions to the problems mentioned in Section 2.3.3. It may be possible to use a two-dimensional version of the CIP metric that would have as inputs the binary masks with ones in the segmented pixels and zeros elsewhere. This would also solve the problem of multiple regions presented by large clutter scenes (and poor segmentations). Additionally, instead of measuring peakiness, entropy may provide better results.

Future experiments should also include large clutter scenes, since these are more representative of SAR data collected in the real world. The challenges associated with using these large clutter scenes are numerous: their size alone requires powerful computer resources and much time for human segmentation. There are also questions about how to run the metrics on the multiple shapes that are sure to be in these larger images. Do we run the metrics on each shape individually, or do we take them as a whole in some way? This question is not problematic for the Percent Pixels Same metric, but for the Partial Directed Hausdorff and Complex Inner Product metrics it must be answered. Another problem with these large images is the number of classes present; there may be trees, shadows, grass, buildings, roads, etc. Statistical differences between some of these classes may be small, which limits the effectiveness of statistical algorithms.

5.4 Thesis Contributions

A particular contribution of this thesis is the implementation of the UM algorithm tested here, which learned the prior probabilities by using the smoothed posterior probabilities from the previous iteration of the same image. This idea was presented by Haker et al., but in their implementation the iterations were not conducted on the same image, but rather on multiple SAR images taken of the same scene. However, SAR images are not produced this way in real time, so the extension of the idea to one single image has been made here. This implementation does offer improvement over the implementation used by Haker et al. for a single image, since the best parameterization of the UM algorithm used six iterations, as opposed to one iteration, which characterizes the performance of the Haker et al. implementation.

Another contribution is the form of the CIP metric used here (see Section 2.3.3), which satisfies the constraint $0 \le m_{CIP} \le 1$ but, unlike previous methods (also discussed in Section 2.3.3), does not discard amplitude information to do so.

However, the primary contribution is the establishment of testing standards for SAR image segmentation algorithms. Valid future tests must use a common data set, master segmentations, and multiple metrics, and must determine surfaces of constant goodness for algorithm-to-algorithm comparisons. Furthermore, determining whether the performance of an algorithm is sufficient to replace human segmentors requires calculation of the Human Threshold and any algorithm not meeting the Human Threshold for all metrics is not sufficient. The goal of this thesis has been achieved: the Statistical Curve Evolution based AT algorithm is the best of the tested algorithms, although it is not good enough to replace human segmentors.

Since objective knowledge about algorithm performance is essential for algorithm improvement, the testing standards established here have the potential to greatly enhance the field of SAR image segmentation.

Appendix

This appendix shows the images which comprise the data set, the master segmentations, sample segmentations, and the complete metric results.

The Data Set Images

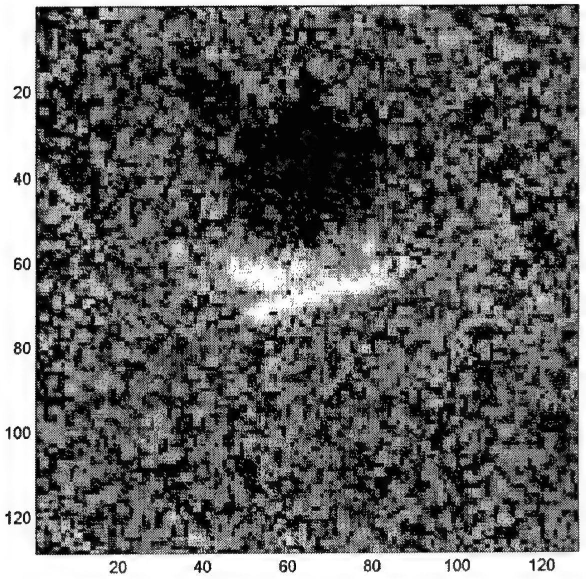

Figure 17—hb03796

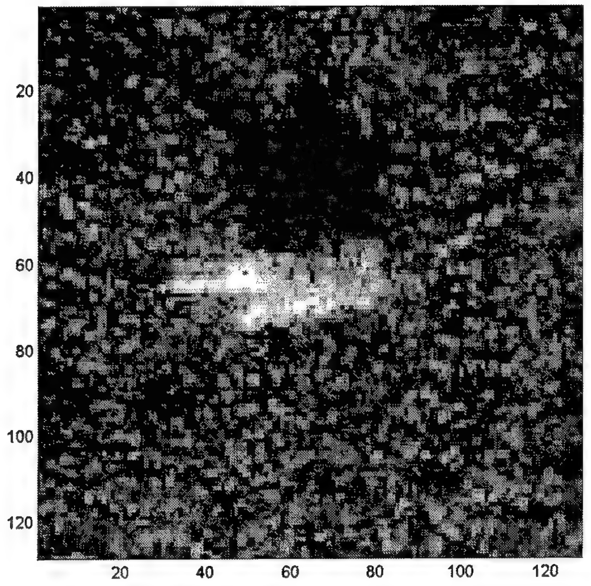

Figure 18—hb03797

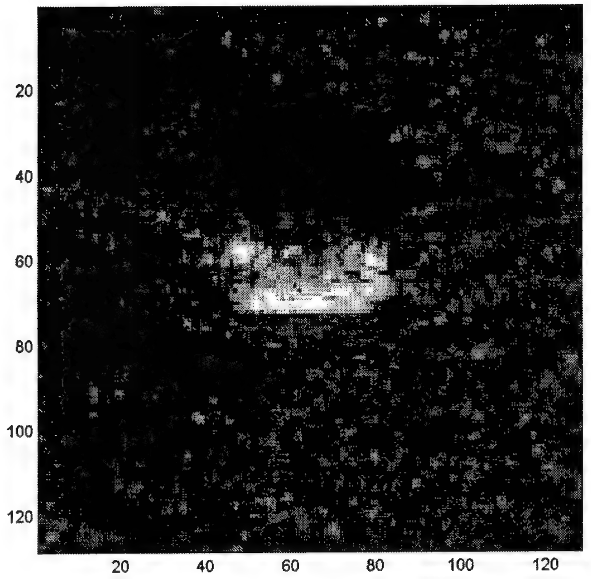

Figure 19—hb03798

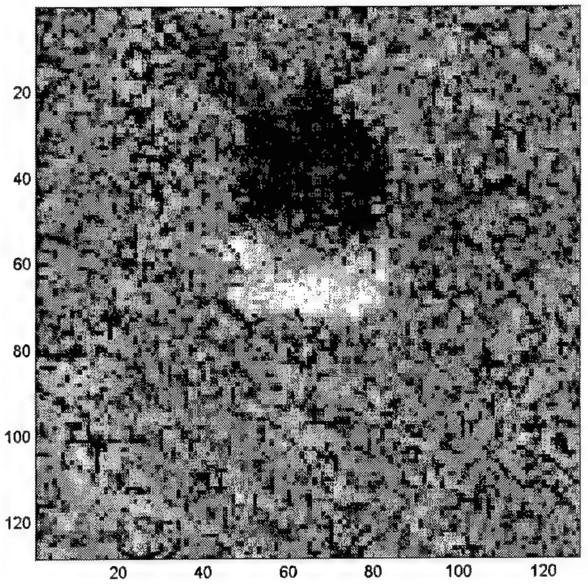

Figure 20—hb03799

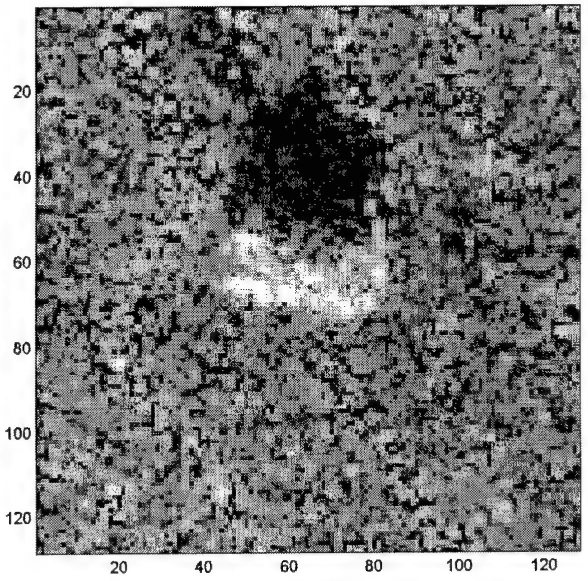

Figure 21—hb03800

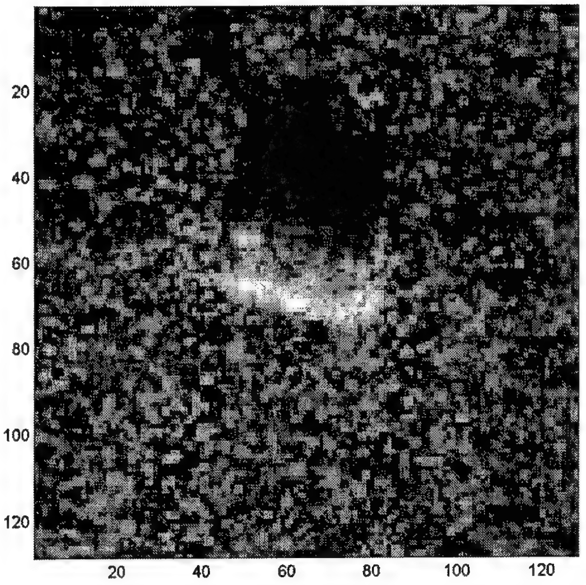

Figure 22—hb03801

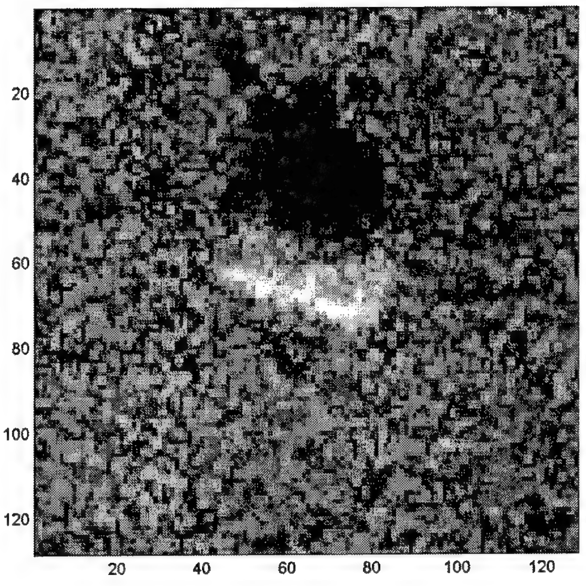

Figure 23—hb03802

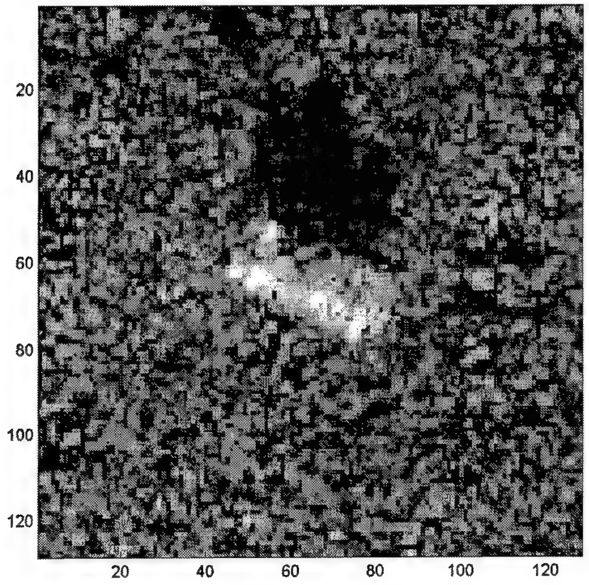

Figure 24—hb03803

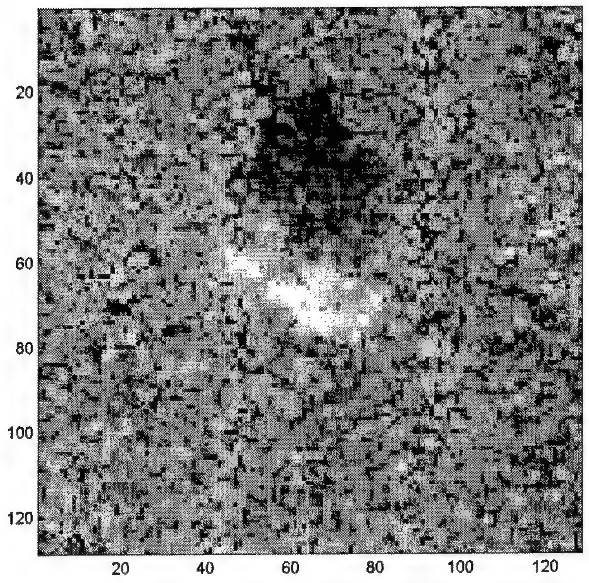

Figure 25—hb03804

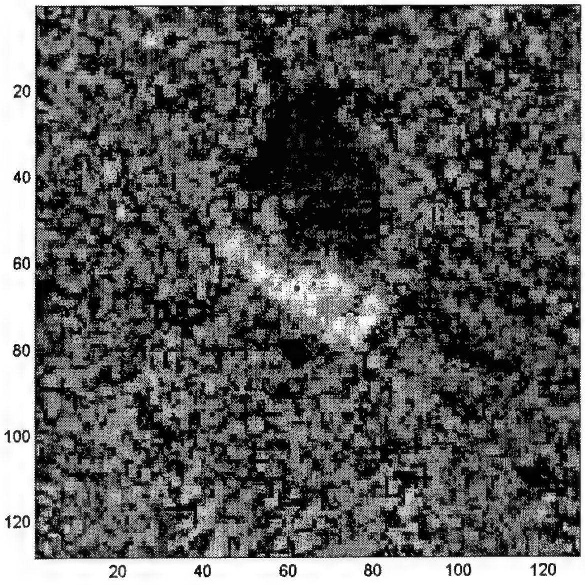

Figure 26—hb03805

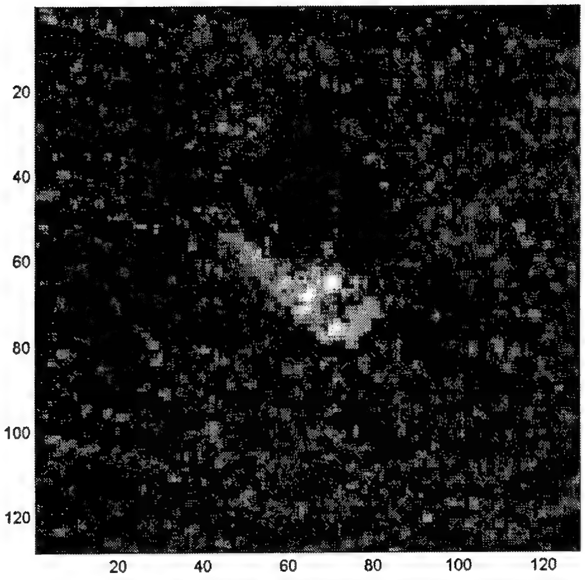

Figure 27—hb03806

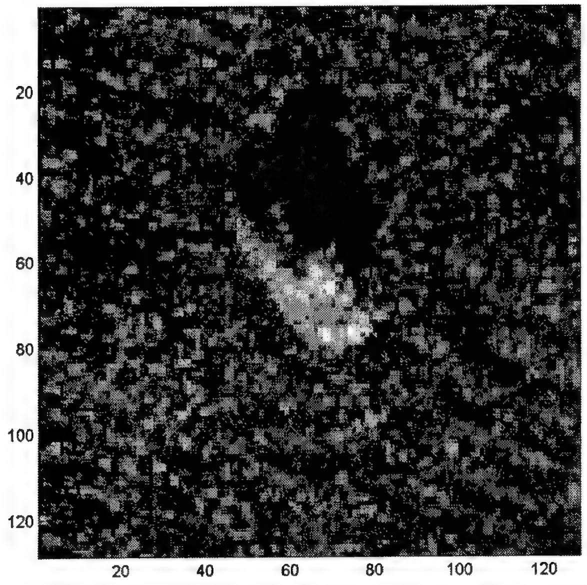

Figure 28—hb03807

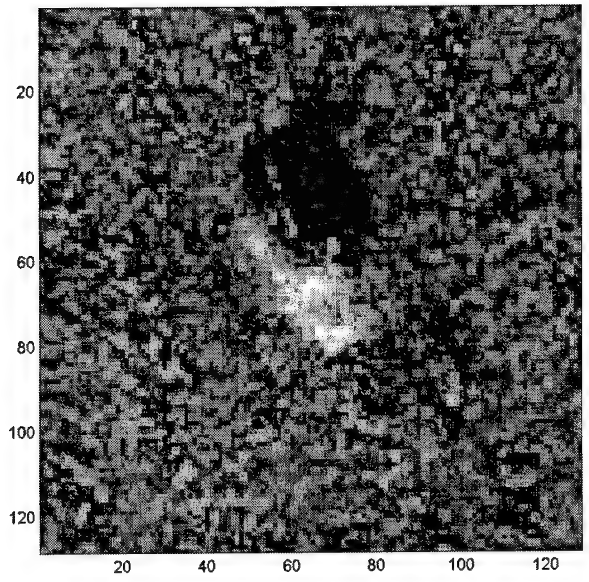

Figure 29—hb03808

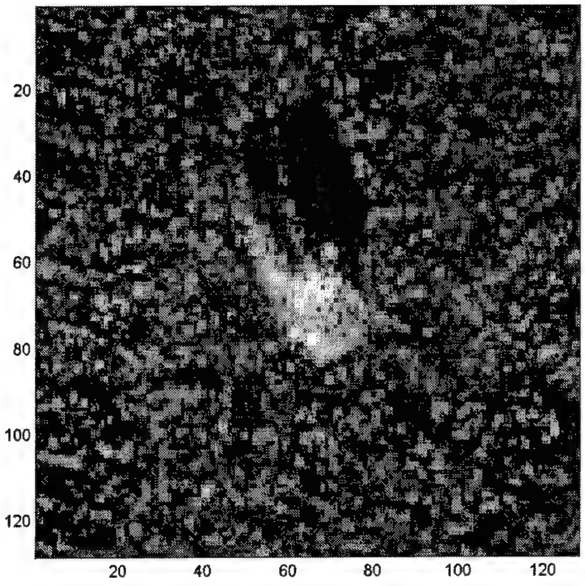

Figure 30—hb03809

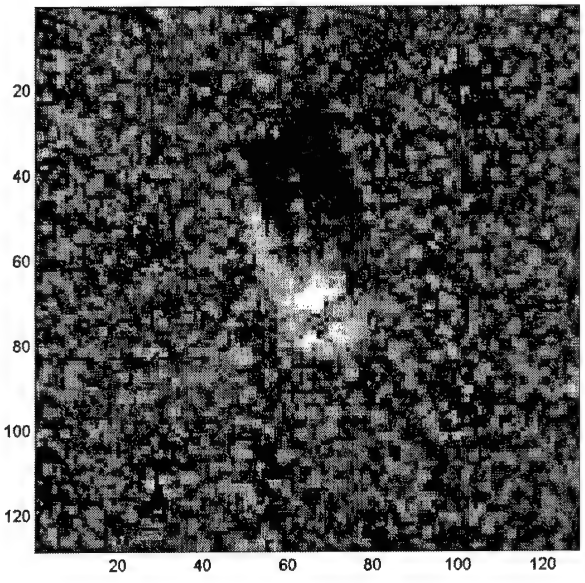

Figure 31—hb03810

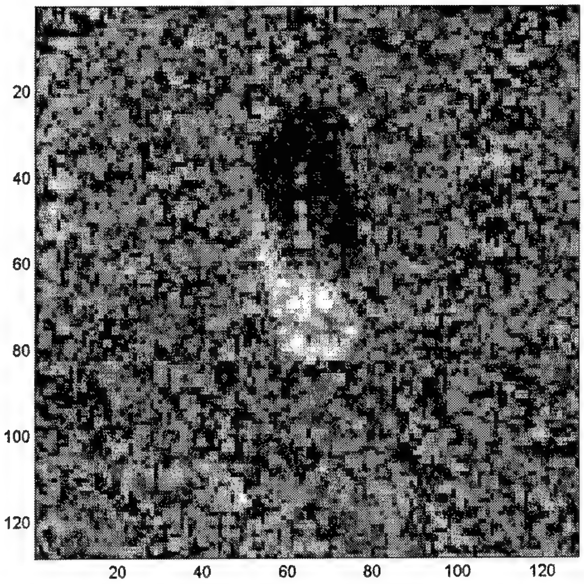

Figure 32—hb03811

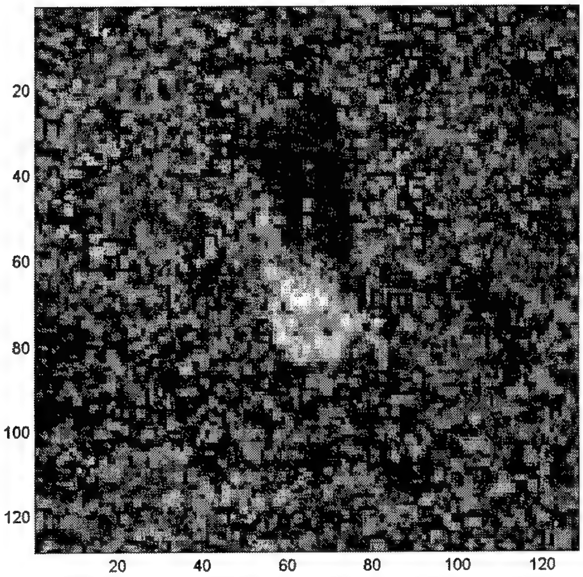

Figure 33—hb03812

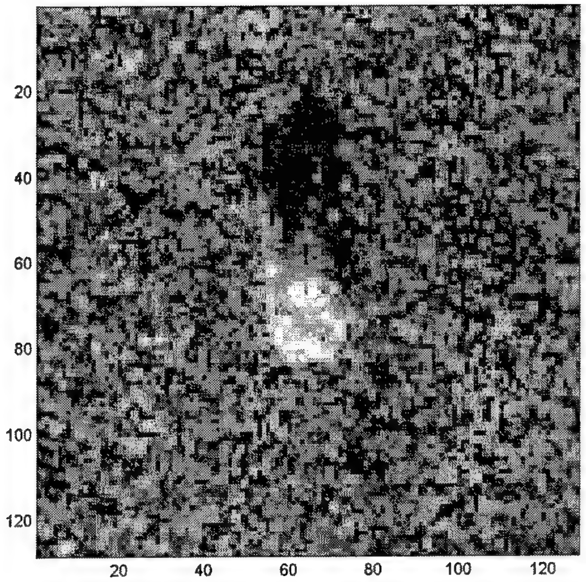

Figure 34—hb03813

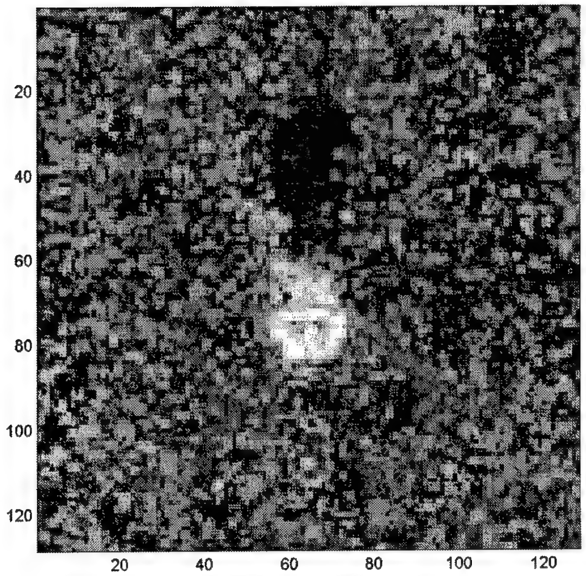

Figure 35—hb03814

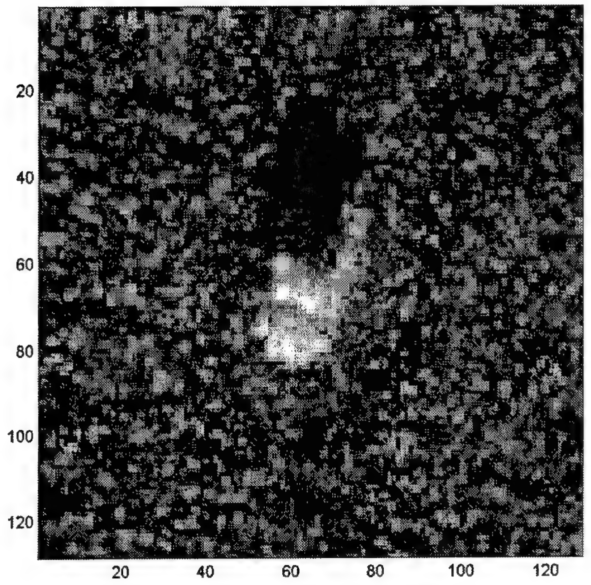

Figure 36—hb03815

The Master Segmentations Images

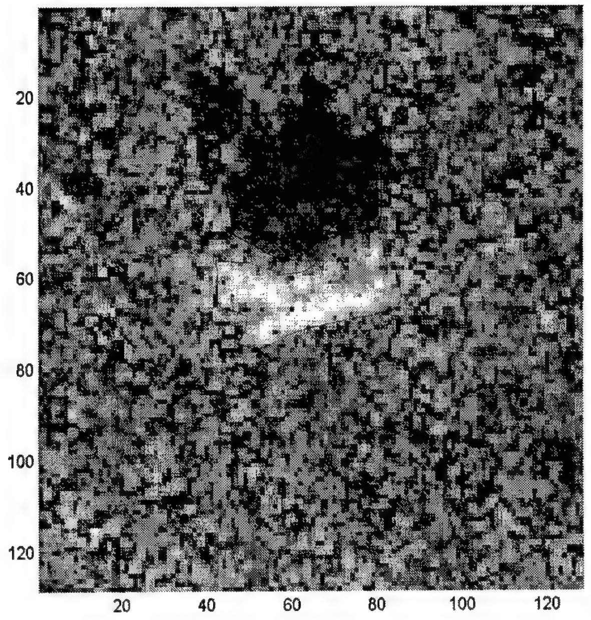

Figure 37—Master segmentation of hb03796

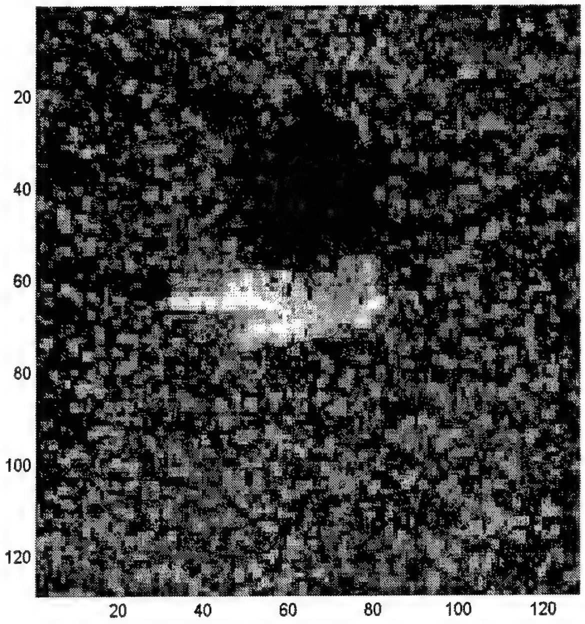

Figure 38—Master segmentation of hb03797

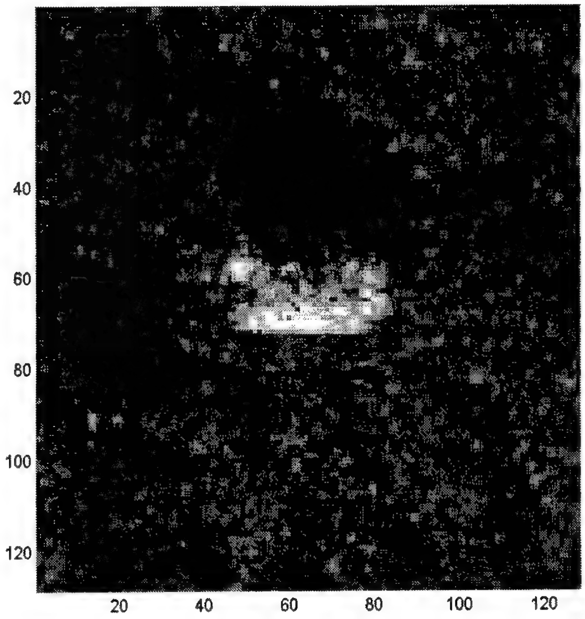

Figure 39—Master segmentation of hb03798

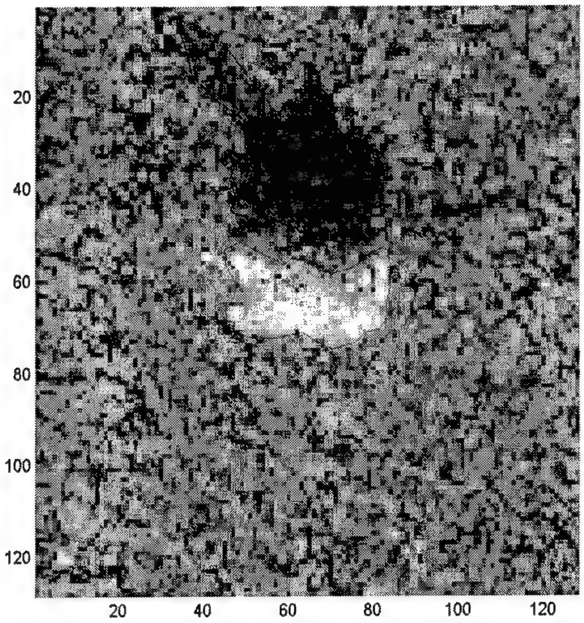

Figure 40—Master segmentation of hb03799

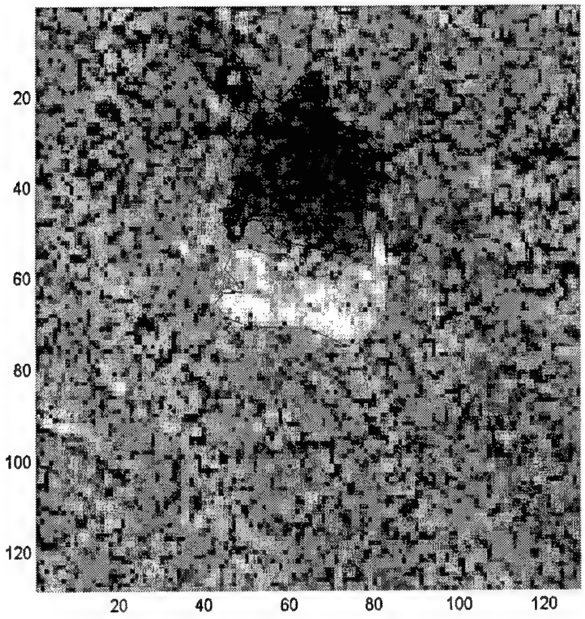

Figure 41—Master segmentation of hb03800

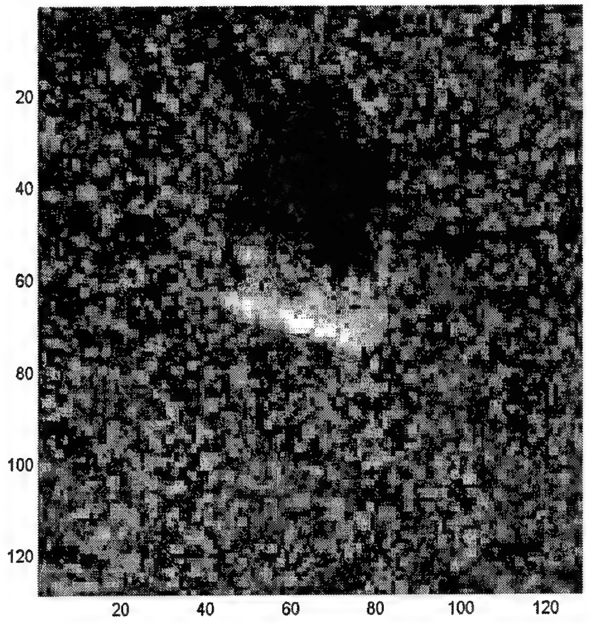

Figure 42—Master segmentation of hb03801

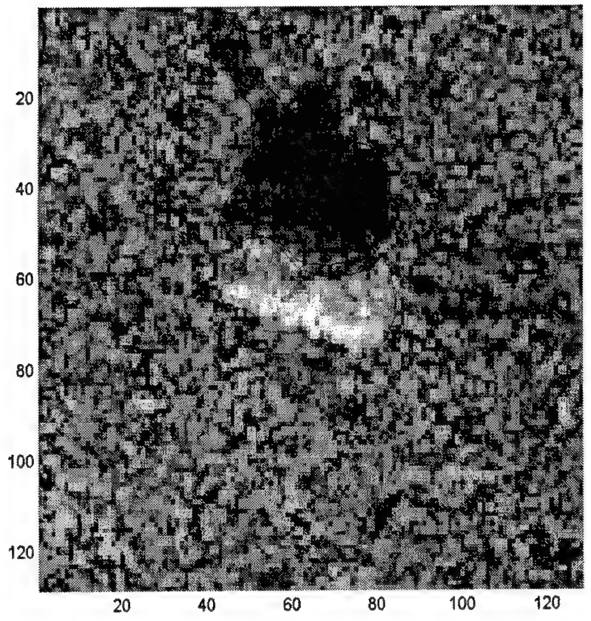

Figure 43—Master segmentation of hb03802

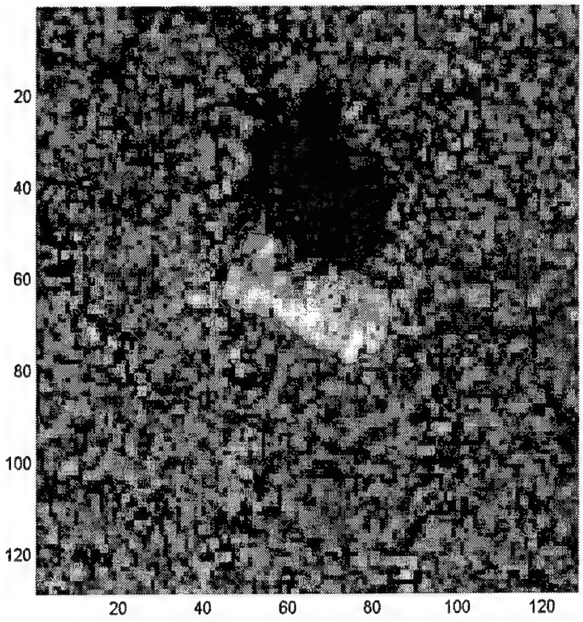

Figure 44—Master segmentation of hb03803

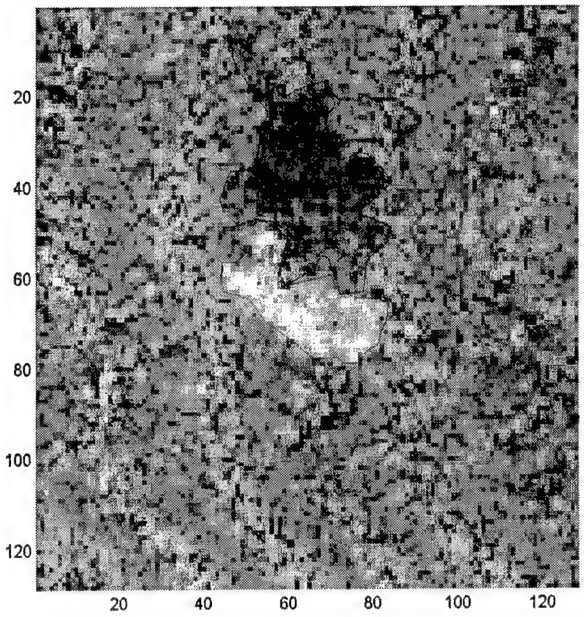

Figure 45—Master segmentation of hb03804

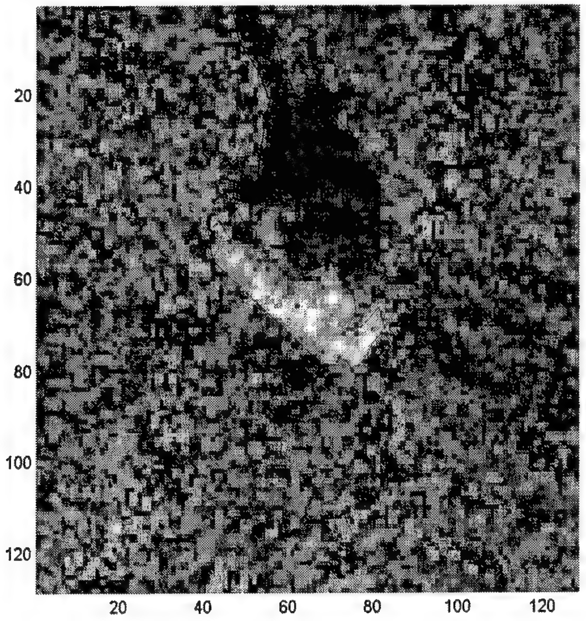

Figure 46—Master segmentation of hb03805

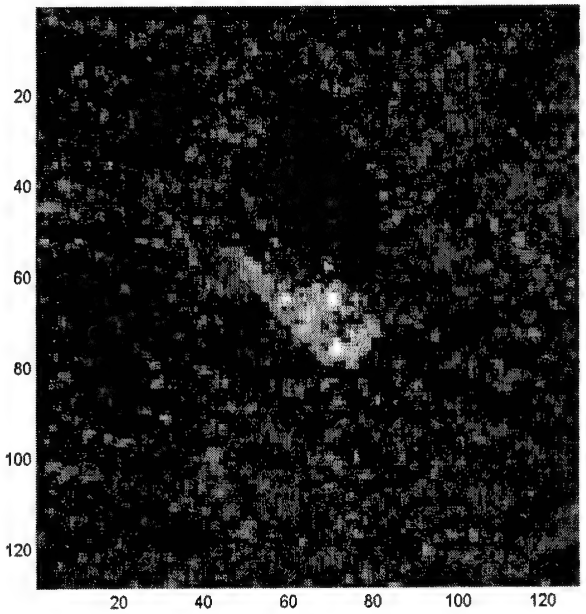

Figure 47—Master segmentation of hb03806

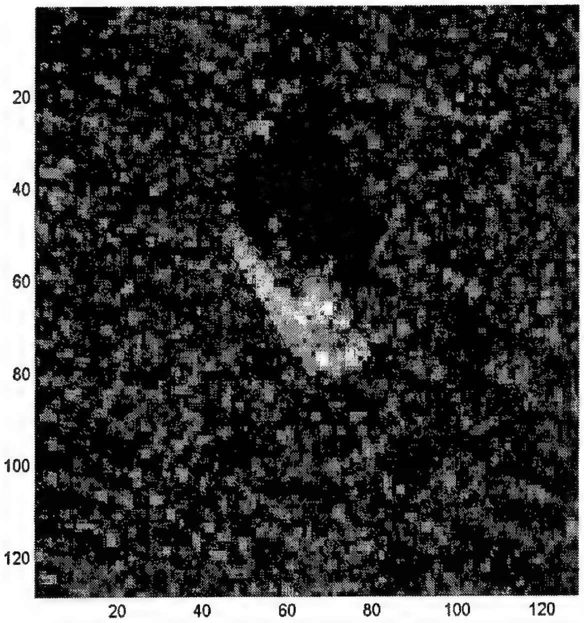

Figure 48—Master segmentation of hb03807

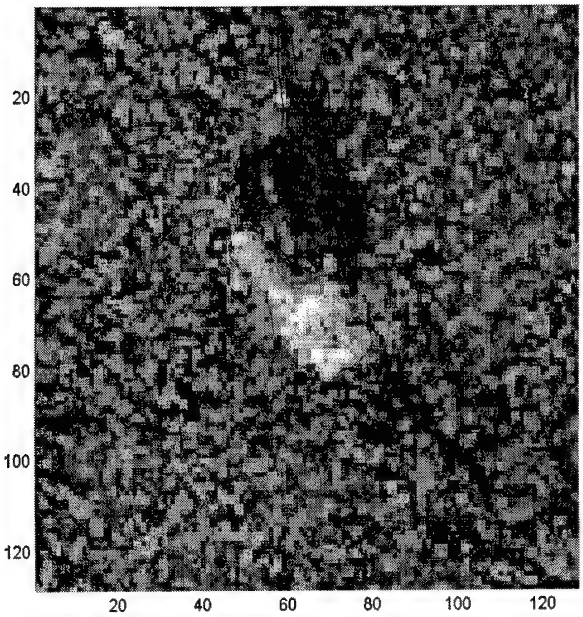

Figure 49—Master segmentation of hb03808

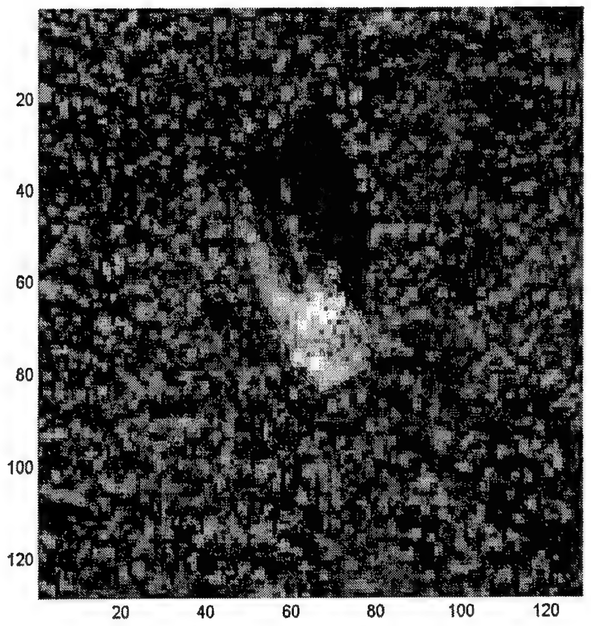

Figure 50—Master segmentation of hb03809

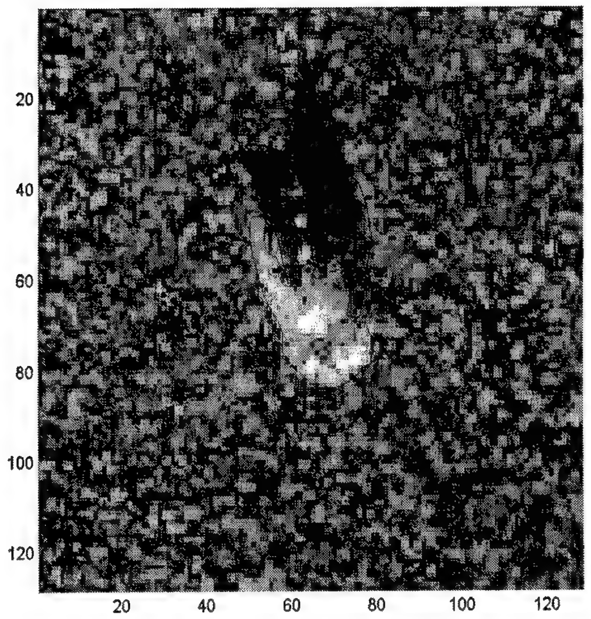

Figure 51—Master segmentation of hb03810

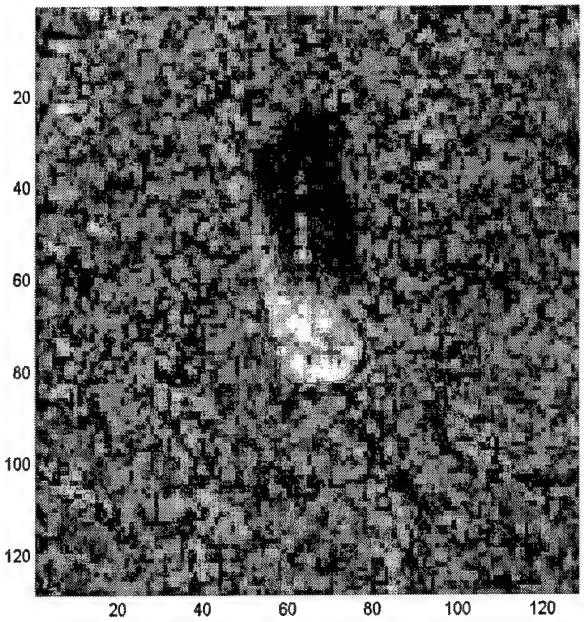

Figure 52—Master segmentation of hb03811

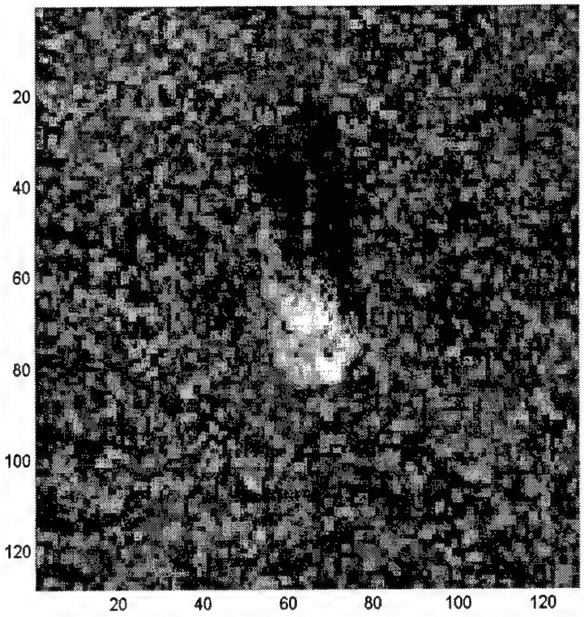

Figure 53—Master segmentation of hb03812

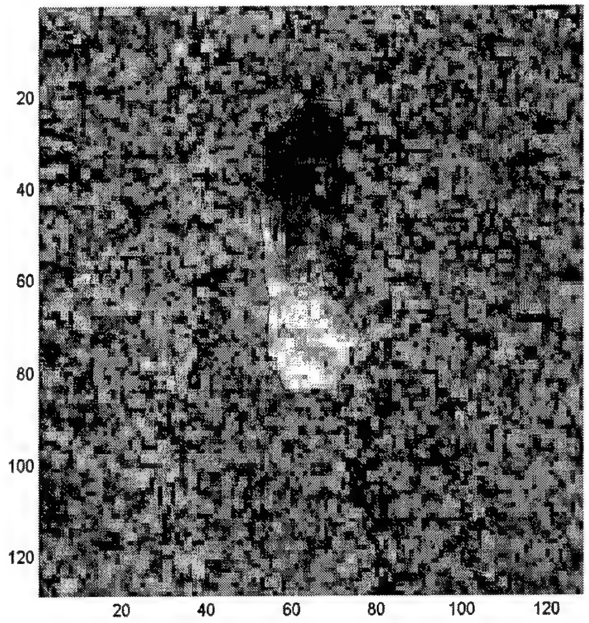

Figure 54—Master segmentation of hb03813

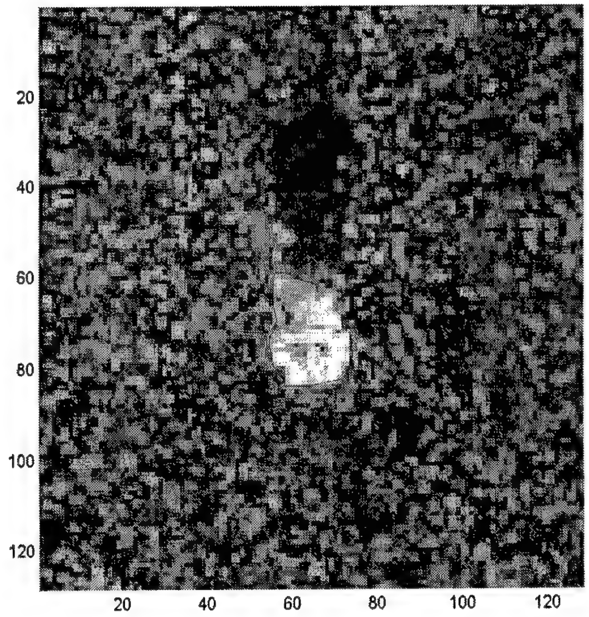

Figure 55—Master segmentation of hb03814

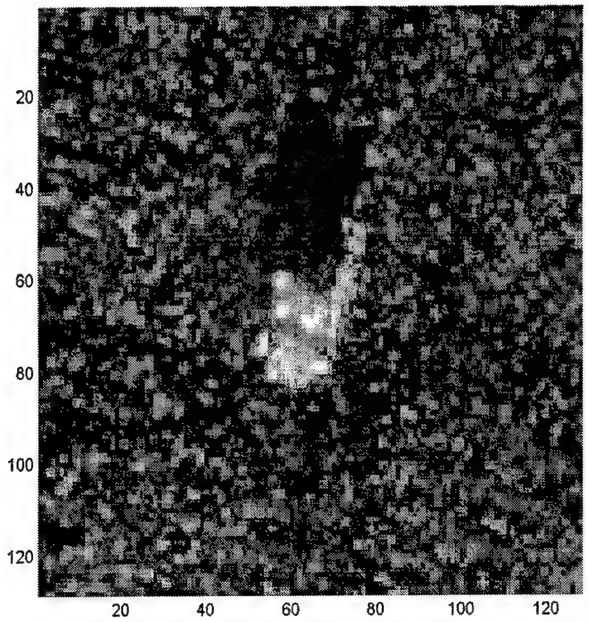

Figure 56—Master segmentation of hb03815

Sample Segmentations—hb03796

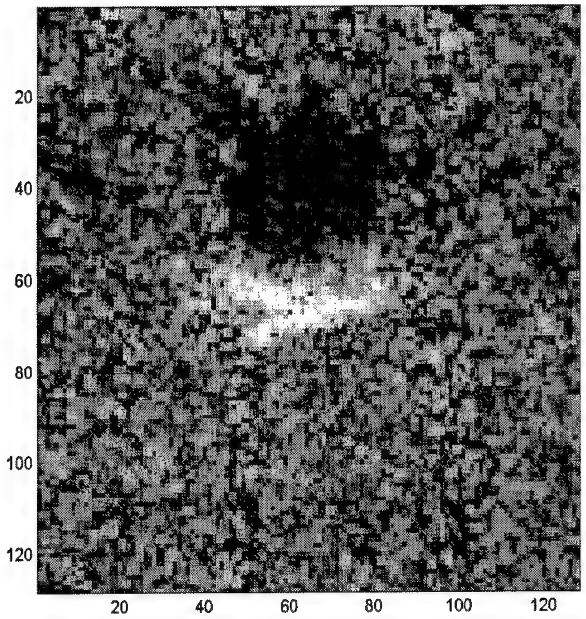

Figure 57—hb03796

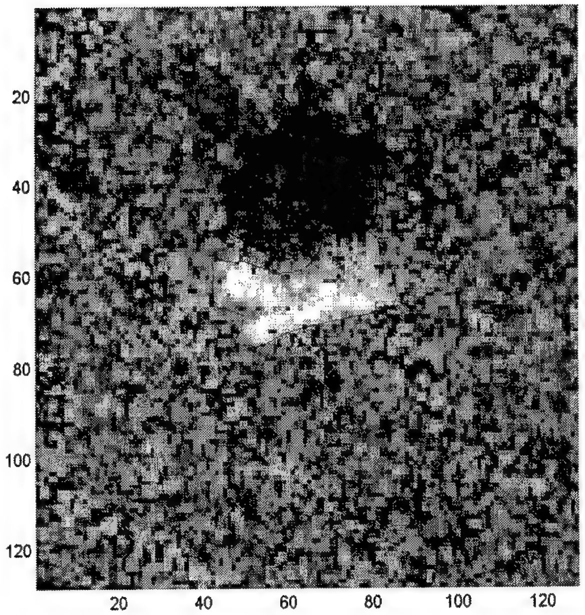

Figure 58—Master segmentation of hb03796

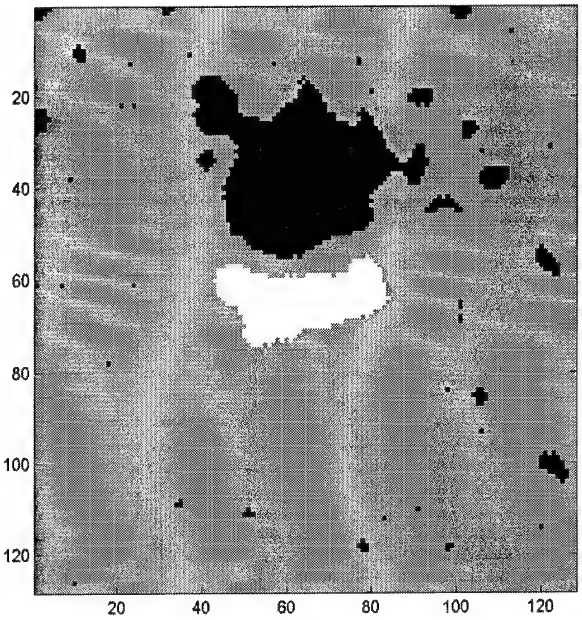

Figure 59—UM run 1 segmentation

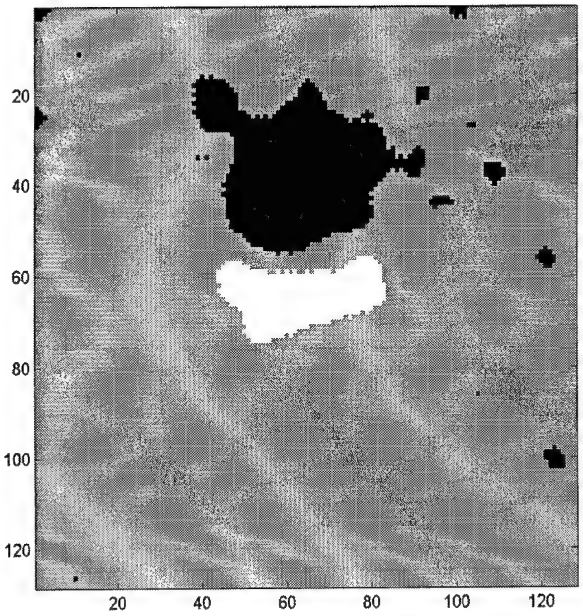

Figure 60—UM run 2 segmentation

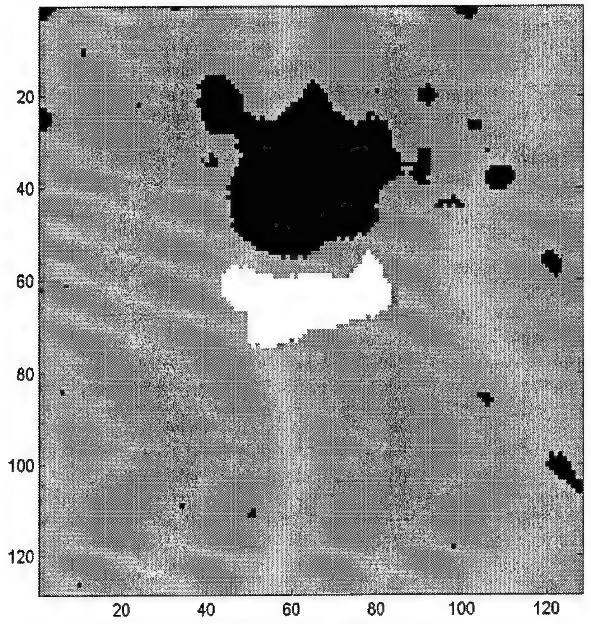

Figure 61—UM run 3 segmentation

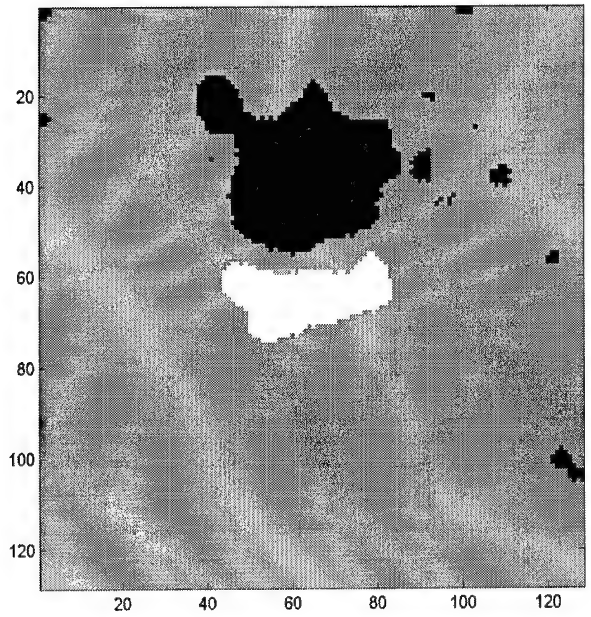

Figure 62—UM run 4 segmentation

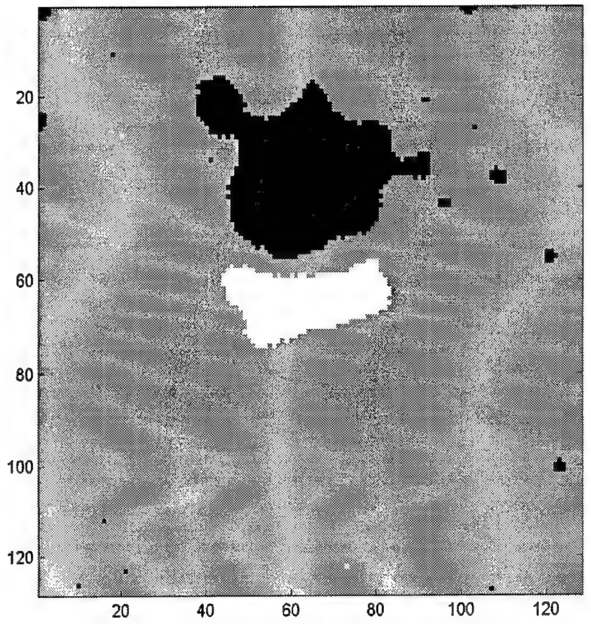

Figure 63—UM run 5 segmentation

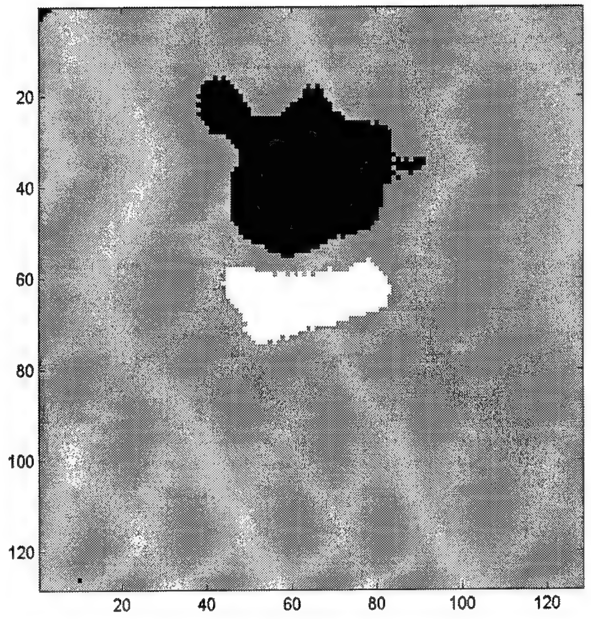

Figure 64—UM run 6 segmentation

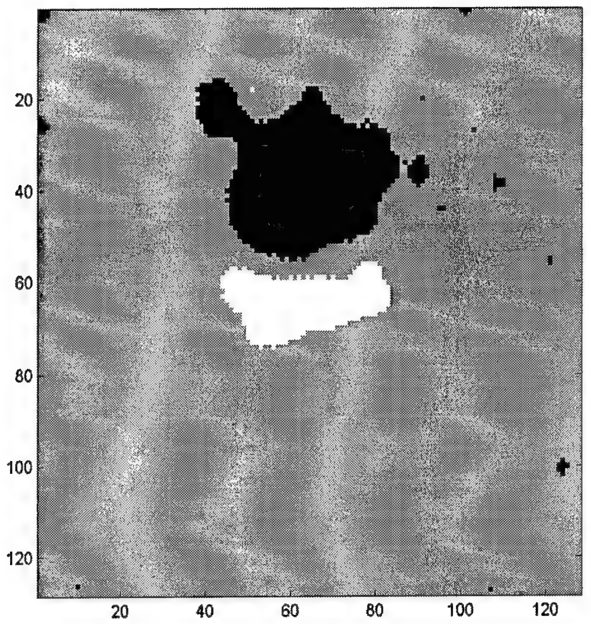

Figure 65—UM run 7 segmentation

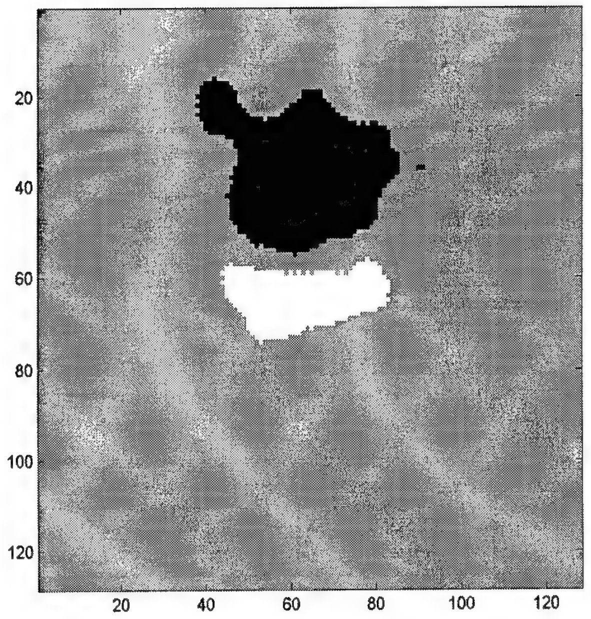

Figure 66—UM run 8 segmentation

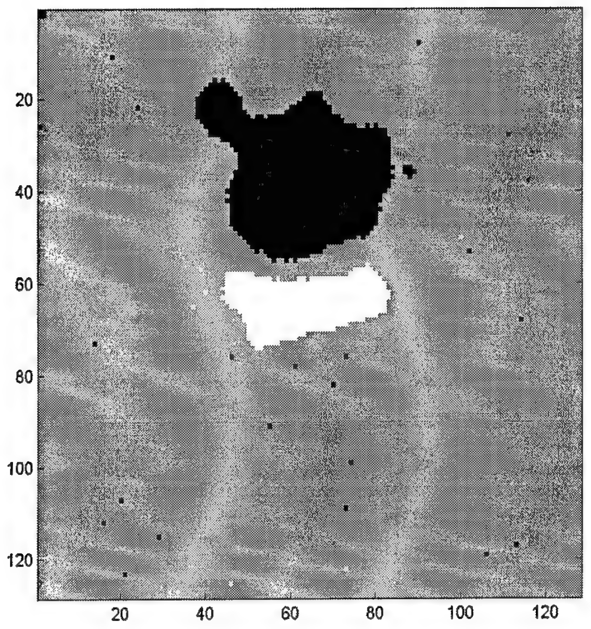

Figure 67—UM run 9 segmentation

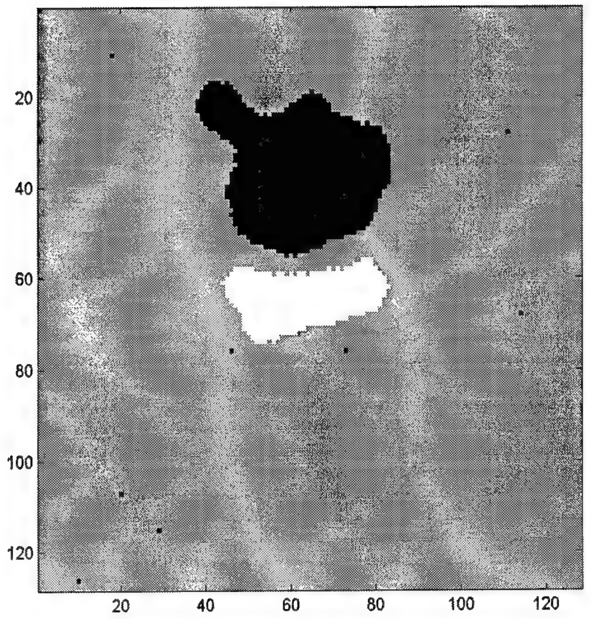

Figure 68—UM run 10 segmentation

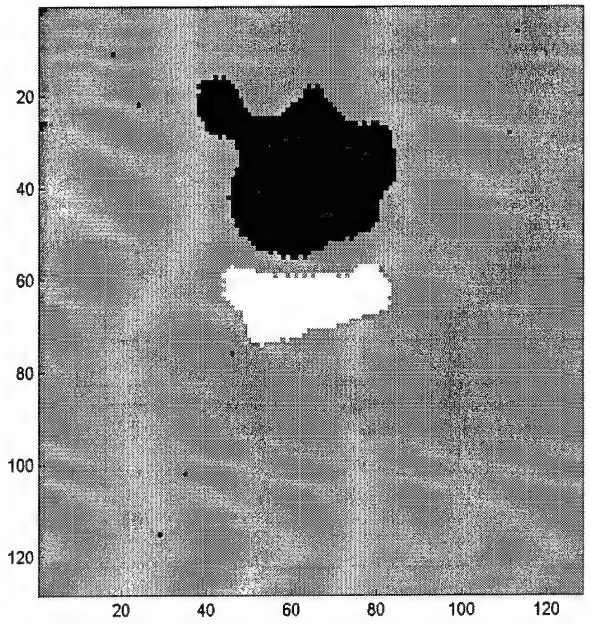

Figure 69—UM run 11 (best overall run) segmentation

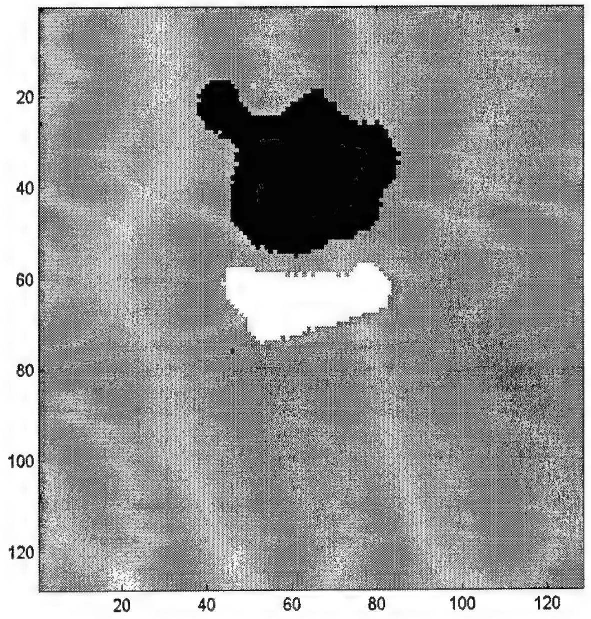

Figure 70—UM run 12 segmentation

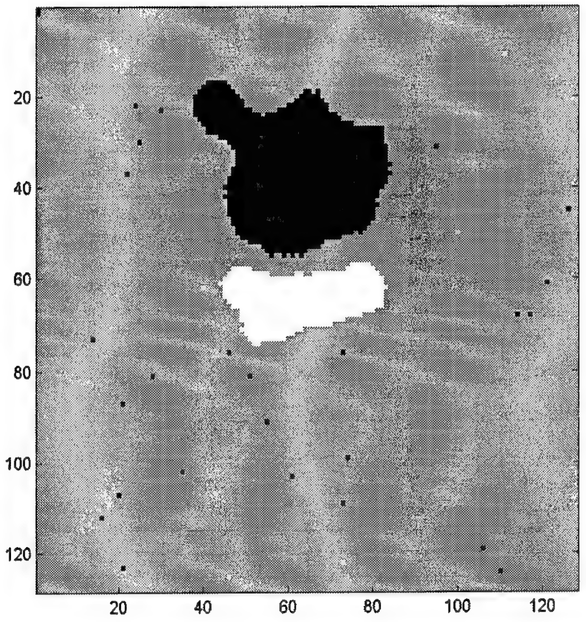

Figure 71—UM run 13 segmentation

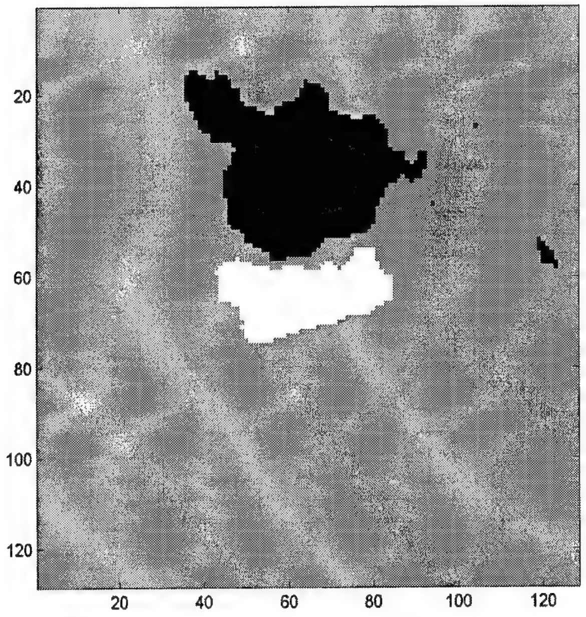

Figure 72—BU run 1 (best overall run) segmentation

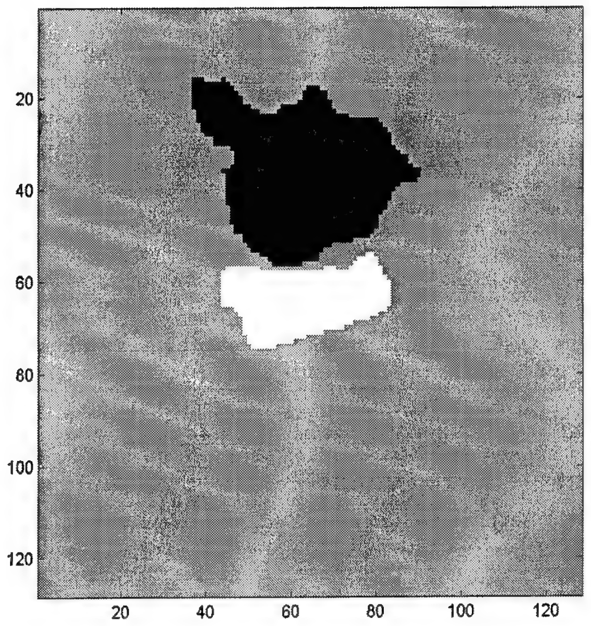

Figure 73—BU run 2 segmentation

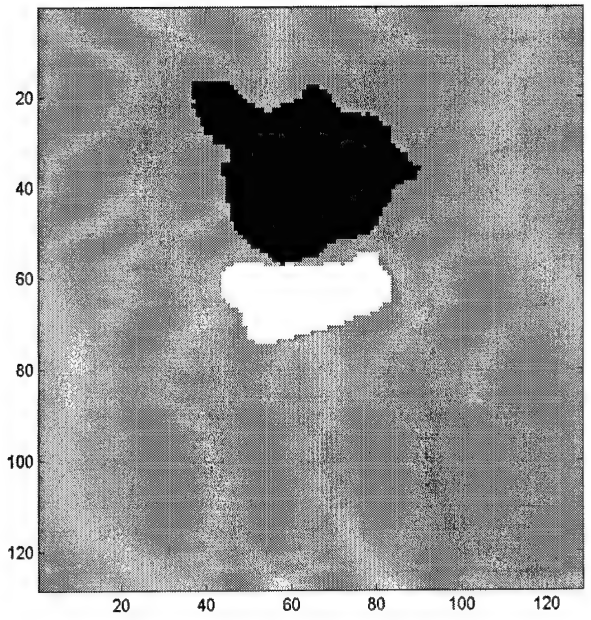

Figure 74—BU run 3 segmentation

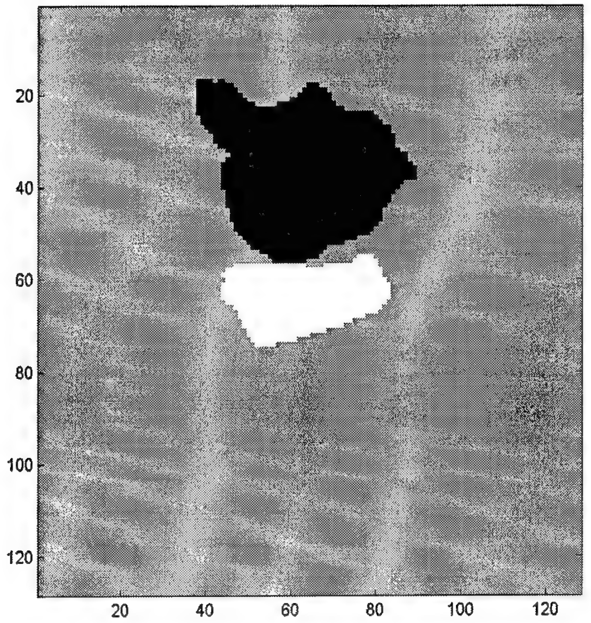

Figure 75—BU run 4 segmentation

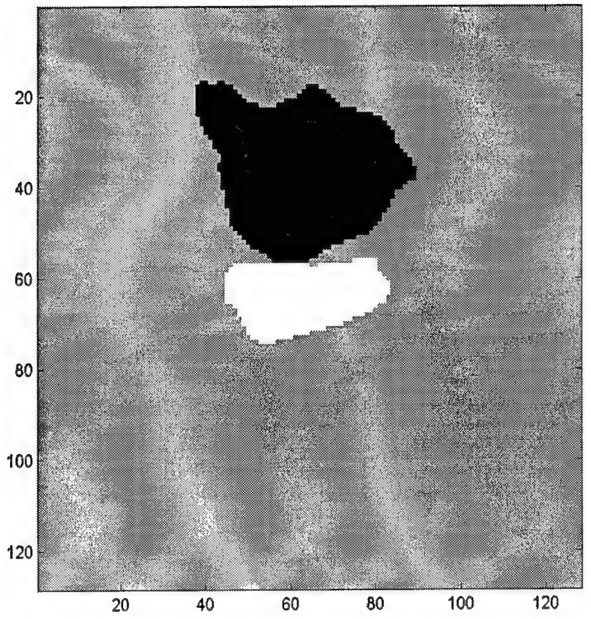

Figure 76—BU run 5 segmentation

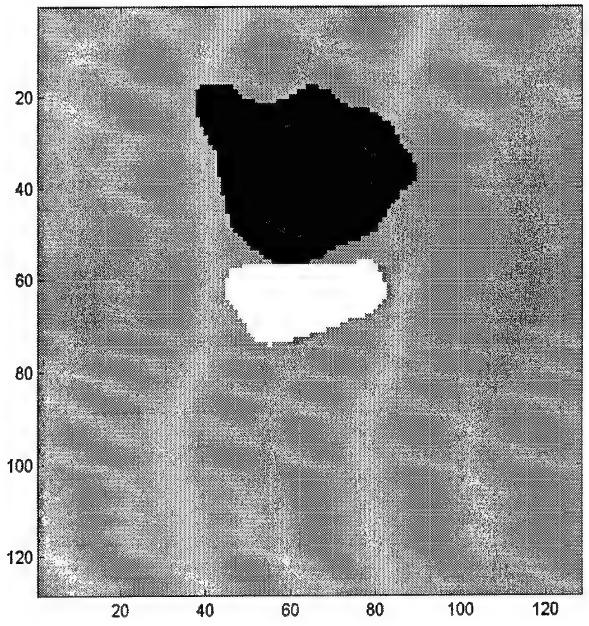

Figure 77—BU run 6 segmentation

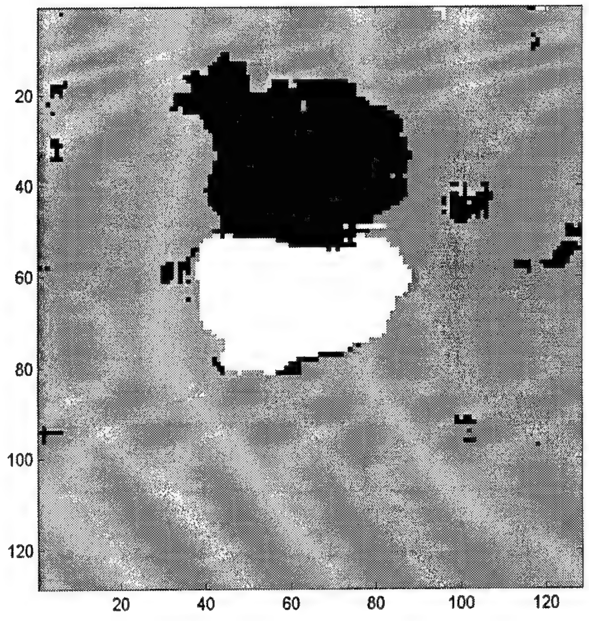

Figure 78—AK run 1 (best overall run) segmentation

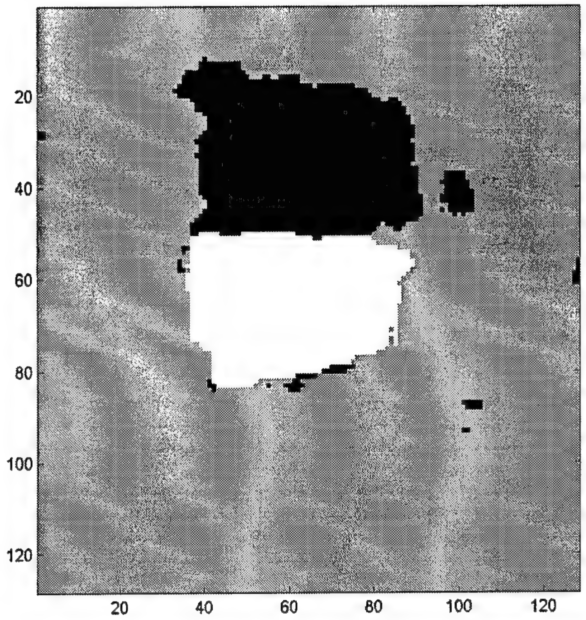

Figure 79—AK run 2 segmentation

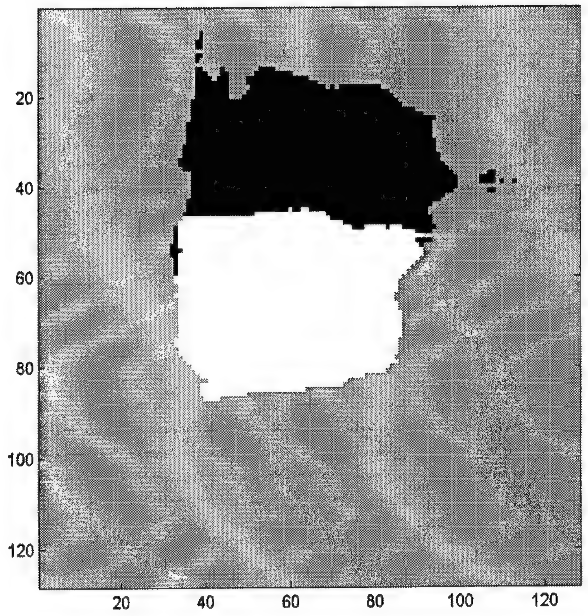

Figure 80—AK run 3 segmentation

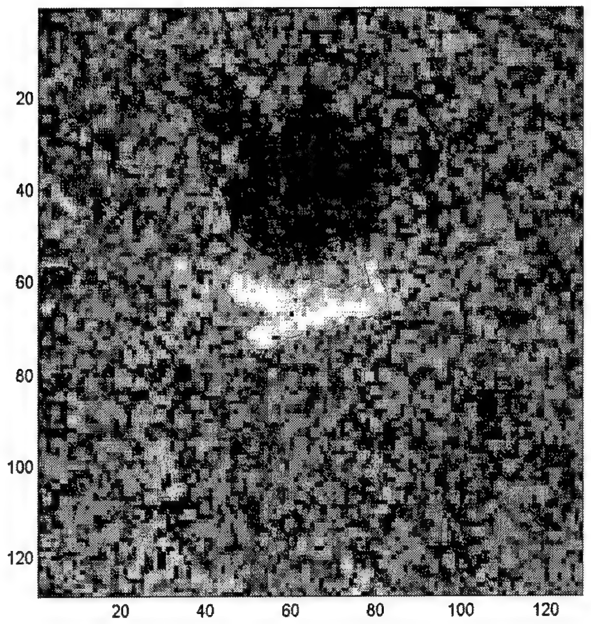

Figure 81—AT run 1 segmentation

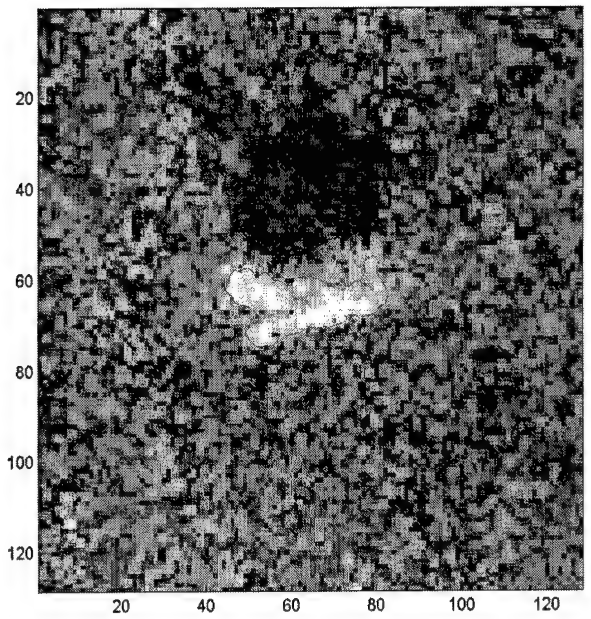

Figure 82—AT run 2 segmentation

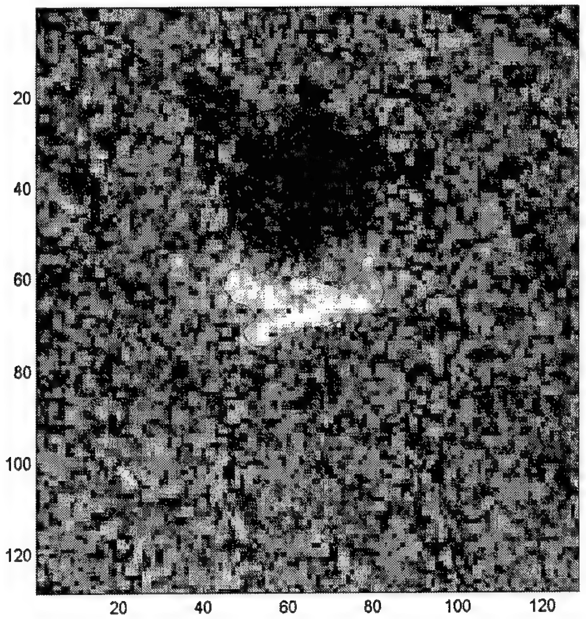

Figure 83—AT run 3 segmentation

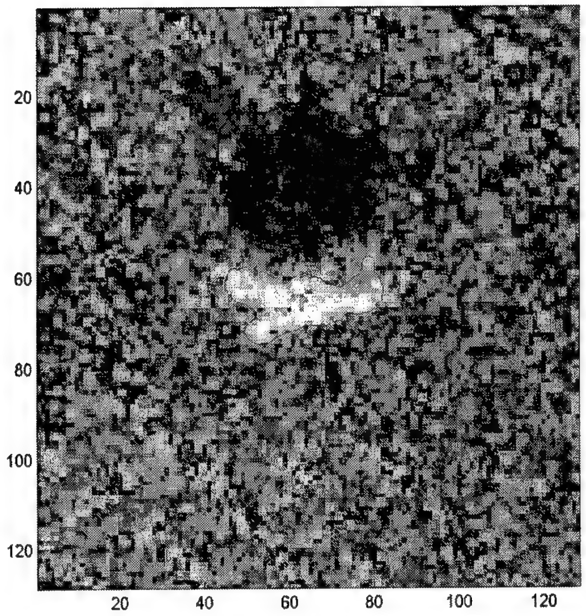

Figure 84—AT run 4 segmentation

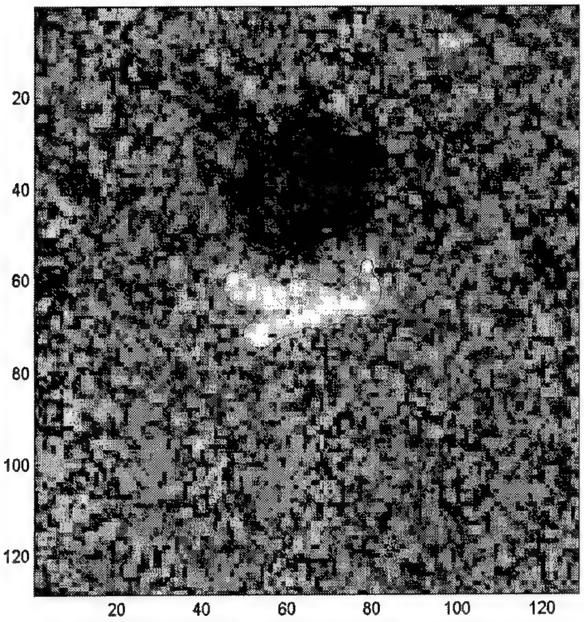

Figure 85—AT run 5 segmentation

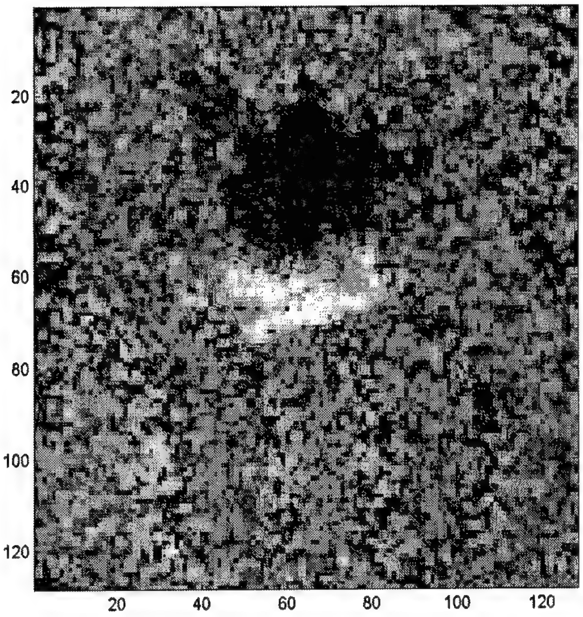

Figure 86—AT run 6 (best overall run) segmentation

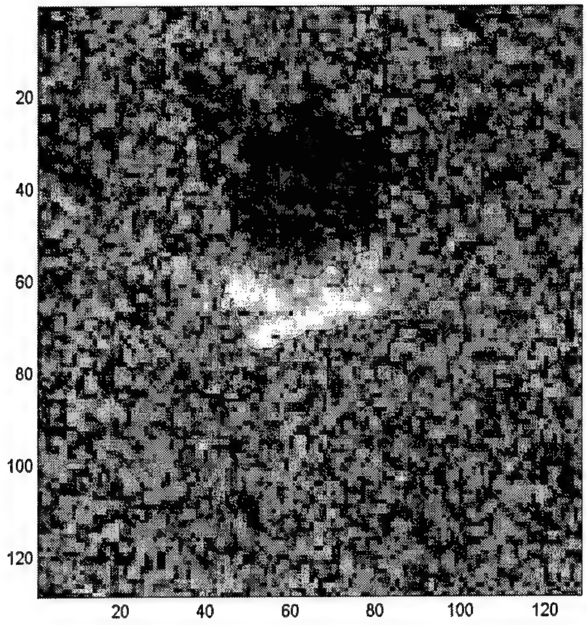

Figure 87—AT run 7 segmentation

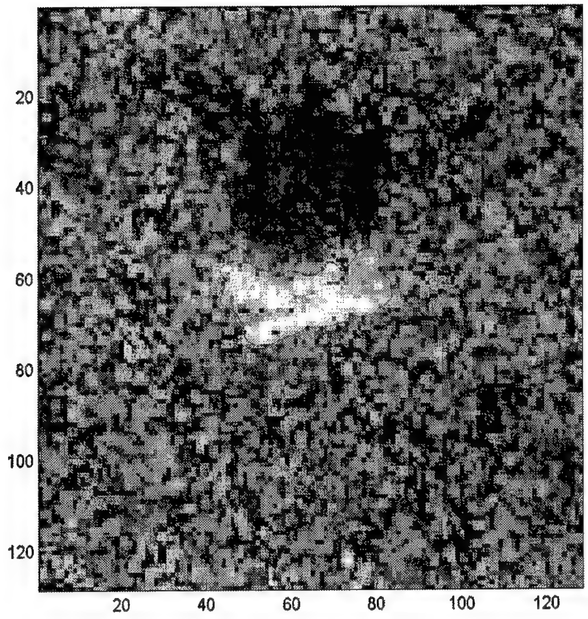

Figure 88—AT run 8 segmentation

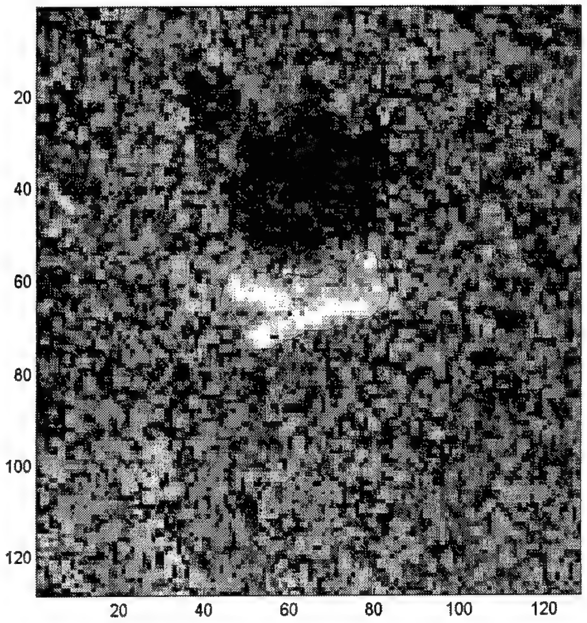

Figure 89—AT run 9 segmentation

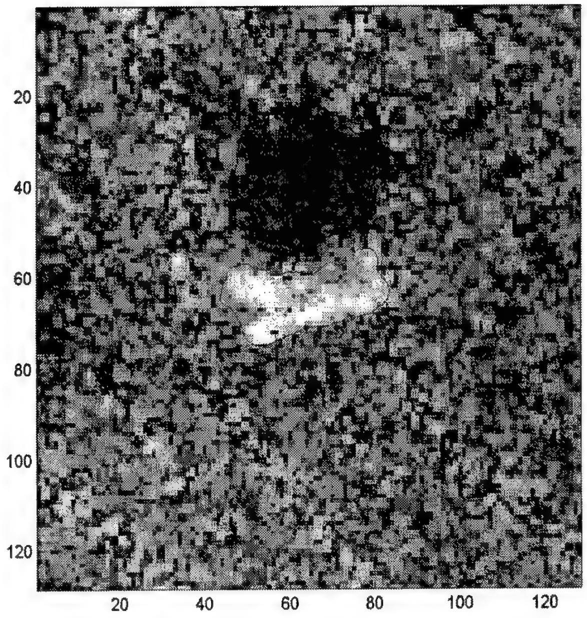

Figure 90—AT run 10 segmentation

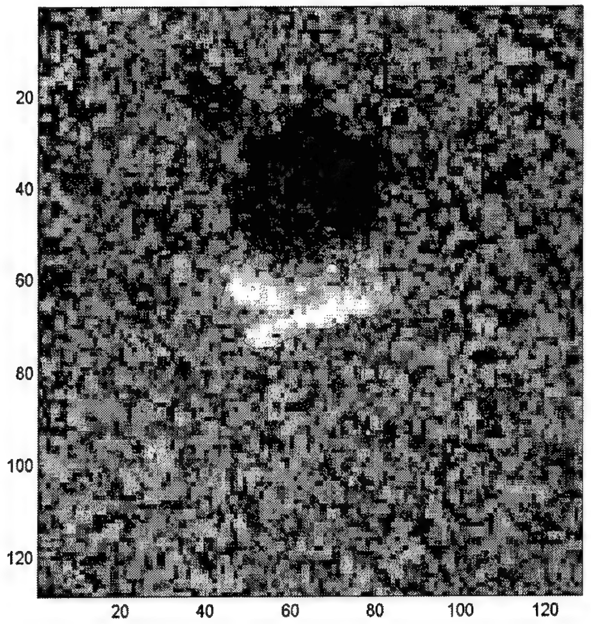

Figure 91—AT run 11 segmentation

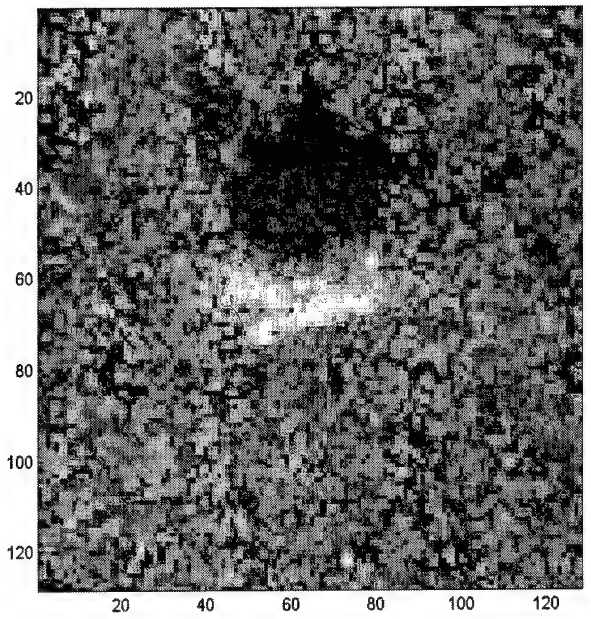

Figure 92—AT run 12 segmentation

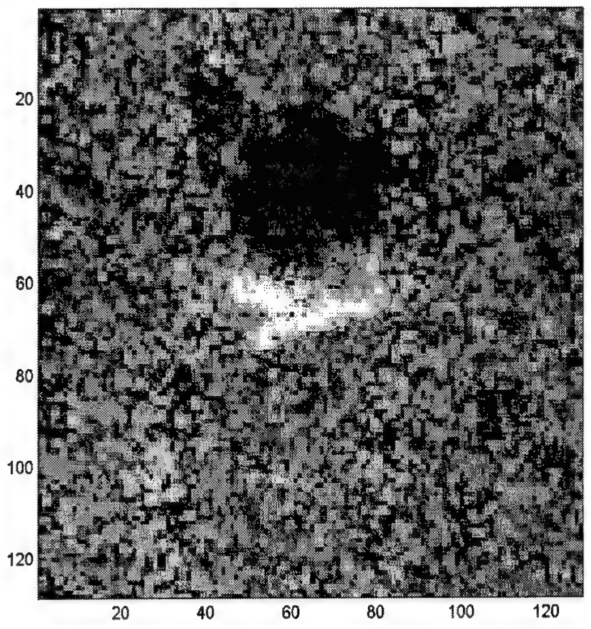

Figure 93—AT run 13 segmentation

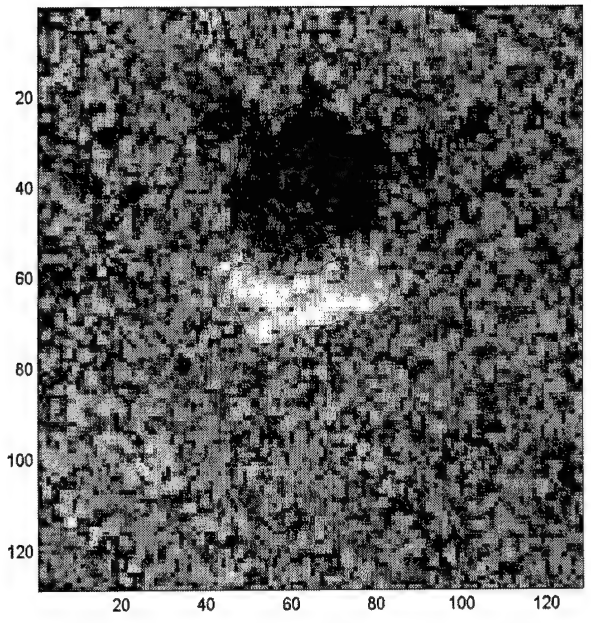

Figure 94—AT run 14 segmentation

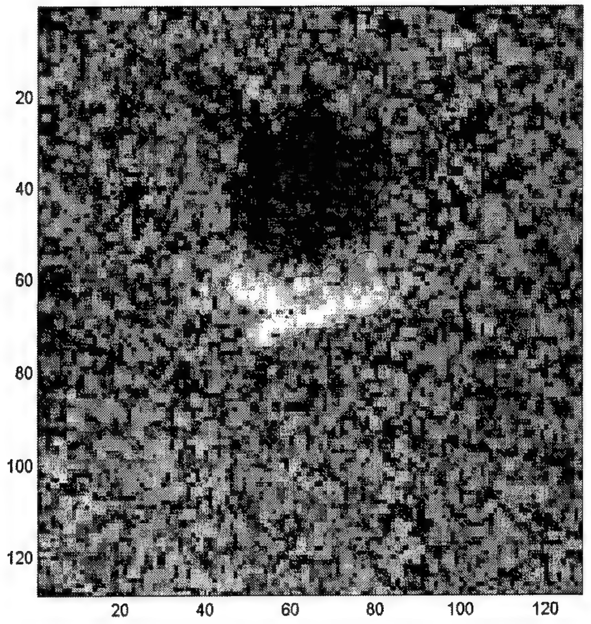

Figure 95—AT run 15 segmentation

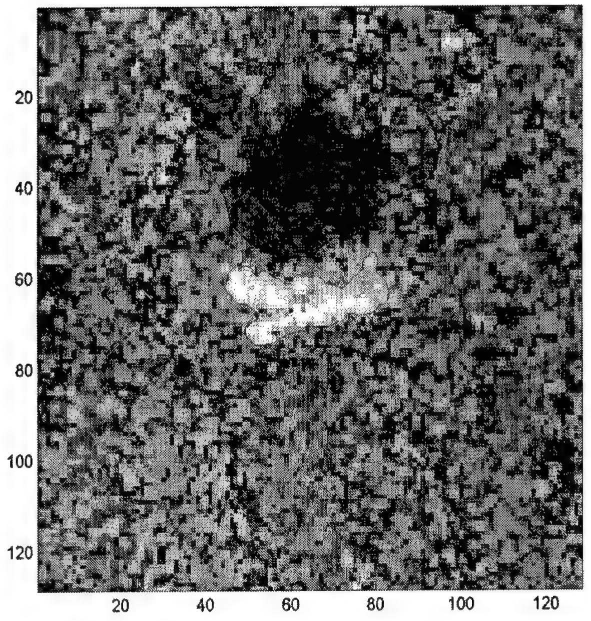

Figure 96—AT run 16 segmentation

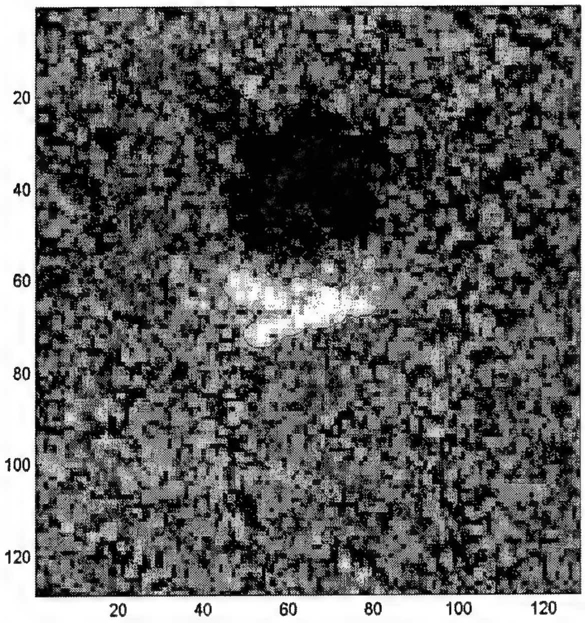

Figure 97—AT run 17 segmentation

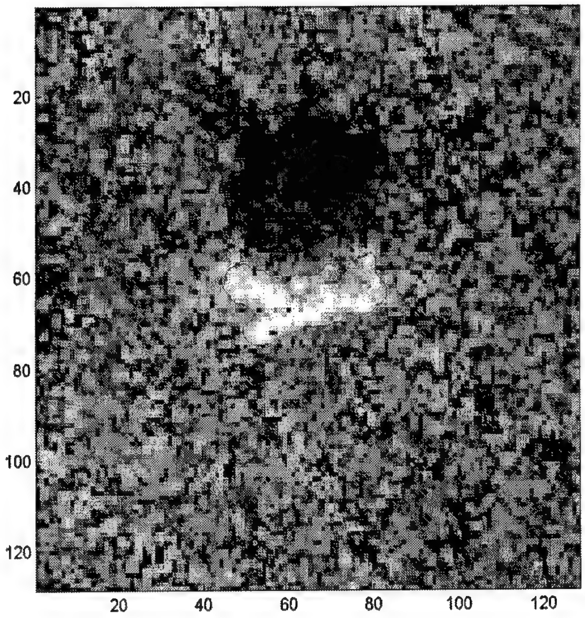

Figure 98—AT run 18 segmentation

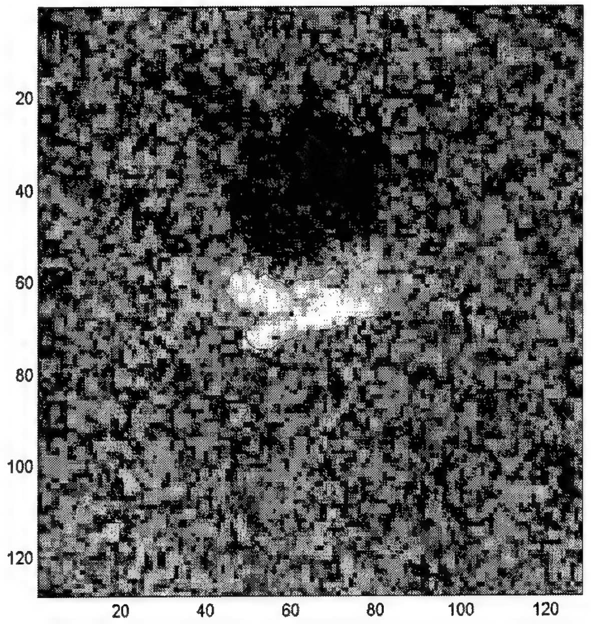

Figure 99—AT run 19 segmentation

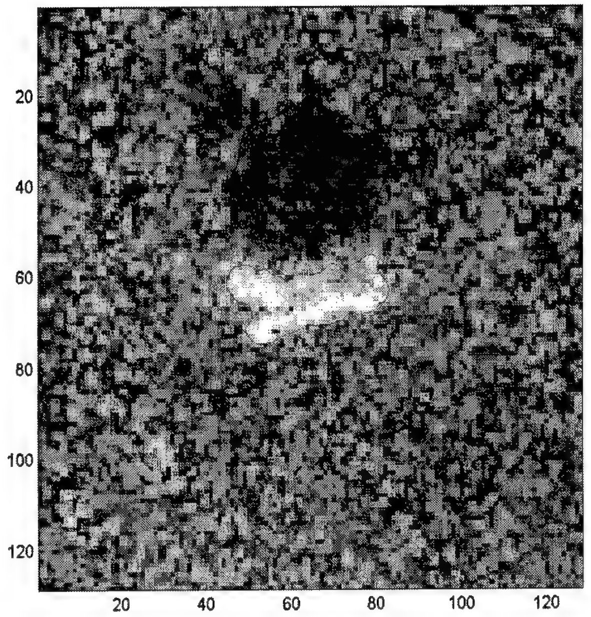

Figure 100—AT run 20 segmentation

Metric results

Table 8—Metric results for all parameterizations of the algorithms

All scores are PPS PDH CIP	Average over 20 images	Meets HT?	Distance from ideal (min is best)	Average of three metrics	Distance from origin
Alg/run					
UM1	0.6410 0.3556 0.5063	No	0.8877	0.5009	0.8908
UM2	0.6763 0.3959 0.5447	No	0.8229	0.5389	0.9543
UM3	0.6569 0.3817 0.5477	No	0.8394	0.5287	0.9365
UM4	0.6828 0.4070 0.5426	No	0.8133	0.5441	0.9624
UM5	0.6886 0.3782 0.5338	No	0.8372	0.5336	0.9499
UM6	0.7067 0.3958 0.5196	No	0.8258	0.5407	0.9623
UM7	0.6908 0.3915 0.5215	No	0.8336	0.5346	0.9499
UM8	0.7054 0.3954 0.5459	No	0.8115	0.5489	0.9756
UM9	0.7082 0.3532 0.5282	No	0.8521	0.5299	0.9515
UM10	0.7196 0.3922 0.5240	No	0.8213	0.5453	0.9728
UM11	0.7066 0.4077 0.5488	No	0.8003	0.5543	0.9832
UM12	0.7148 0.3786 0.5305	No	0.8294	0.5413	0.9673
UM13	0.7166 0.3646 0.5113	No	0.8503	0.5308	0.9528
BU1	0.7552 0.4341 0.5377	No	0.7706	0.5757	1.0237
BU2	0.7621 0.3913 0.5224	No	0.8094	0.5586	1.0035
BU3	0.7535 0.3877 0.5091	No	0.8226	0.5501	0.9885
BU4	0.7477 0.3762 0.5153	No	0.8293	0.5464	0.9829

All scores are PPS PDH CIP	Average over 20 images	Meets HT?	Distance from ideal (min is best)	Average of three metrics	Distance from origin
Alg/run					
BU5	0.7363 0.3534 0.4847	No	0.8679	0.5248	0.9497
BU6	0.7120 0.3234 0.4599	No	0.9124	0.4984	0.9072
AK1	0.5044 0.1070 0.5009	No	1.1367	0.3708	0.7189
AK2	0.4648 0.0896 0.4769	No	1.1785	0.3438	0.6719
AK3	0.3957 0.0767 0.4837	No	1.2183	0.3187	0.6296
AT1	0.6577 0.3854 0.5519	No	0.8341	0.5317	0.9412
AT2	0.7039 0.4255 0.5542	No	0.7852	0.5612	0.9918
AT3	0.6947 0.4256 0.5479	No	0.7922	0.5561	0.9818
AT4	0.6832 0.4364 0.5432	No	0.7917	0.5542	0.9758
AT5	0.6674 0.4324 0.5648	No	0.7888	0.5549	0.9754
AT6	0.7758 0.4917 0.5572	No	0.7104	0.6082	1.0743
AT7	0.7683 0.4931 0.5591	No	0.7106	0.6068	1.0705
AT8	0.7427 0.4799 0.5507	No	0.7339	0.5911	1.0418
AT9	0.7203 0.4819 0.5482	No	0.7421	0.5835	1.0255
AT10	0.7024 0.4692 0.5521	No	0.7555	0.5746	1.0092
AT11	0.7239 0.4648 0.5386	No	0.7587	0.5757	1.0149
AT12	0.7633 0.4745 0.5456	No	0.7339	0.5945	1.0514
AT13	0.7346 0.4671 0.5343	No	0.7559	0.5786	1.0214
AT14	0.7115 0.4700 0.5567	No	0.7487	0.5794	1.0184
AT15	0.6937 0.4635 0.5478	No	0.7656	0.5683	0.9981

All scores are PPS PDH CIP	Average over 20 images	Meets HT?	Distance from ideal (min is best)	Average of three metrics	Distance from origin
Alg/run					
AT16	0.7060 0.4177 0.5478	No	0.7937	0.5572	0.9864
AT17	0.7348 0.4487 0.5498	No	0.7596	0.5778	1.0216
AT18	0.7206 0.4478 0.5279	No	0.7784	0.5654	0.9992
AT19	0.6991 0.4474 0.5335	No	0.7833	0.5600	0.9867
AT20	0.6843 0.4406 0.5516	No	0.7834	0.5588	0.9832

Bibliography

Awwal, Abdul A. S., et al. "Improved Correlation Discrimination Using an Amplitude-Modulated Phase-Only Filter," Applied Optics 29-2: 233-236 (January 1990).

Beauchemin, M. et al. "On the Hausdorff Distance Used for the Evaluation of Segmentation Results," Canadian Journal of Remote Sensing 24-1: 3-8 (March 1998).

Defense Advanced Research Projects Agency. MSTAR (Public) Data Set. Wright-Patterson AFB. (http://www.mbvlab.wpafb.af.mil/public/MBVDATA)

Department of the Air Force. Air Force Basic Doctrine. AFDD-1. Maxwell AFB: 1997.

Haker, Steven et al. "Knowledge-Based Segmentation of SAR Data with Learned Priors," IEEE Transactions on Image Processing (not yet published)

Hoover, Adam et al. "An Experimental Comparison of Range Image Segmentation Algorithms," IEEE Transactions on Pattern Analysis and Machine Intelligence 18-7: 673-679 (July 1996).

Kim, Andrew and Krim, Hamid. "Hierarchical Stochastic Modeling of SAR Imagery for Segmentation/Compression," IEEE Transactions on Signal Processing 47-2: 458-468 (February 1999).

Kuttikkad, Shyam and Chellappa, Rama. "Statistical Modeling and Analysis of High-Resolution Synthetic Aperture Radar Images," Statistics and Computing 10: 133-145 (2000).

Power, Gregory J. "Multimetric Inter-Algorithmic Evaluation of SAR Segmentation Algorithms," Proceedings of SPIE 4382 (April 2001).

----- and Awwal, Abdul A.S. "Optoelectronic Complex Inner Product for Evaluating Quality of Image Segmentation," Proceedings of SPIE 4114 (July 2000).

Stremler, Ferrel G. Introduction to Communication Systems (3rd Edition). Addison-Wesley Publishing Company, Inc., 1990.

Weisenseel, Robert A. et al. "Markov Random Field Segmentation Methods for SAR Target Chips," Proceedings of SPIE 3721 (1999).

Yezzi, Anthony, Jr. et al. "A Fully Global Approach to Image Segmentation via Coupled Curve Evolution Equations," Journal of Visual Communication and Image Representation (not yet published)

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 20-03-2001	**2. REPORT TYPE** Master's Thesis	**3. DATES COVERED** May 2000-March 2001

4. TITLE AND SUBTITLE

AN OBJECTIVE EVALUATION OF FOUR SAR IMAGE SEGMENTATION ALGORITHMS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Gregga, Jason, B, Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Air Force Institute of Technology, Graduate School of Engineering and Management
2950 P Street, Bldg 640
WPAFB, OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GE/ENG/01M-12

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Gregory J. Power, PhD
AFRL/SNAT
2241 Avionics Circle, Bldg 620
WPAFB, OH 45433
DSN 785-1115x4366

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Because of the large number of SAR images the Air Force generates and the dwindling number of available human analysts, automated methods must be developed. A key step towards automated SAR image analysis is image segmentation. There are many segmentation algorithms, but they have not been tested on a common set of images, and there are no standard test methods. This thesis evaluates four SAR image segmentation algorithms by running them on a common set of data and objectively comparing them to each other and to human segmentors. This objective comparison uses a multi-metric approach with a set of master segmentations as ground truth. The metric results are compared to a Human Threshold, which defines the performance of human segmentors compared to the master segmentations. Also, methods that use the multi-metrics to determine the best algorithm are developed. These methods show that of the four algorithms, Statistical Curve Evolution produces the best segmentations; however, none of the algorithms are superior to human segmentors. Thus, with the Human Threshold and Statistical Curve Evolution as benchmarks, this thesis establishes a new and practical framework for testing SAR image segmentation algorithms.

15. SUBJECT TERMS

SAR image segmentation, algorithm evaluation, segmentation metrics, multi-metrics

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	18. NUMBER OF PAGES	**19a. NAME OF RESPONSIBLE PERSON** Steven C. Gustafson, AFIT/ENG
a. REPORT	b. ABSTRACT	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(include area code)*
U	U	U	UU	123	(937)255-3636x4598